

分布定数線路の周波数特性

愛媛大学工学部

8531037m

祖父江匠真

1 はじめに

分布定数線路の周波数特性を調べた。

2 分布定数線路の周波数特性

図 1 の同軸ケーブルの仕様表より, 3D-2V ケーブルの減衰特性の折れ線グラフを図 2 に示す。

・ 同軸ケーブルの仕様 (参考: 藤倉電線資料)

引用 URL

- http://www.oriixrentec.co.jp/tmsite/knownow_doujiku53.html
- <http://home3.highway.ne.jp/welcome/tv/cable/syurui.htm>

名称	外径 (mm)	特性インピーダンス (Ω)	静電容量 (nF/km)	波長短縮率 (%)	減衰特性 (dB/km)		
					1MHz	10MHz	200MHz
1.5C-2V	2.9	75 \pm 3	67	67	27	82	390
3C-2V	5.6	75 \pm 3	67	67	12	40	195
5C-2V	7.5	75 \pm 3	67	67	7.6	25	125
1.5D-2V	2.9	50 \pm 2	100	67	27	85	420
3D-2V	5.5	50 \pm 2	100	67	13	44	220
5D-2V	7.5	50 \pm 2	100	67	7.3	26	125
8D-2V	11.5	50 \pm 2	100	67	4.8	17	85
RG58/U	5.0	53.5	94	67	13	42	200
RG58A/U	5.0	50	102	67	14	48	230

図 1: 同軸ケーブルの仕様

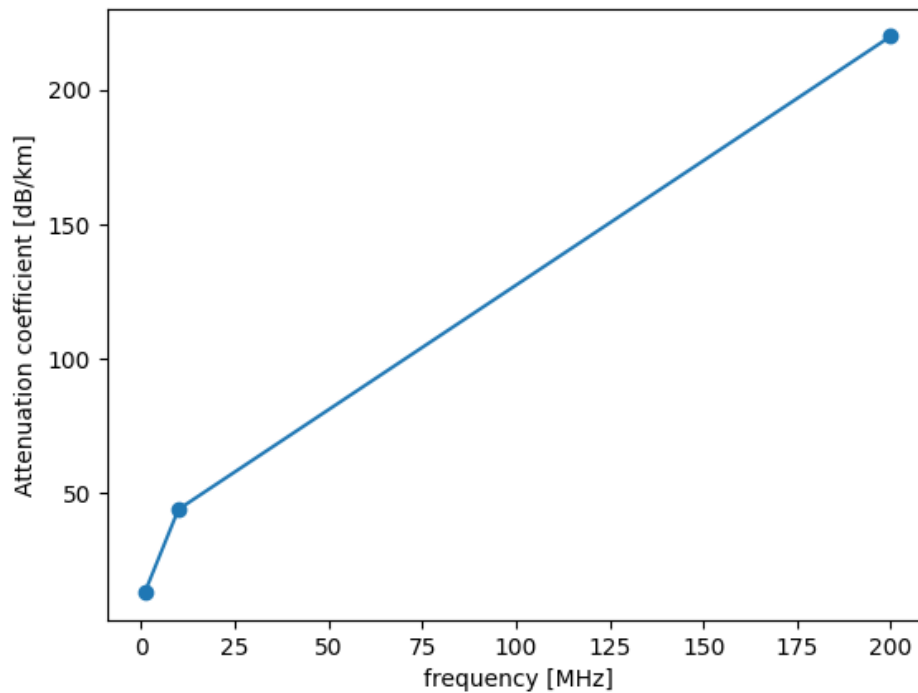


図 2: 減衰特性の折れ線グラフ

伝搬定数 ($= \alpha + j\beta$) における減衰定数 α については, 図 2 の周波数に対応した値を計算に用いた.

図 3 の回路図について, 周波数特性を調べる.

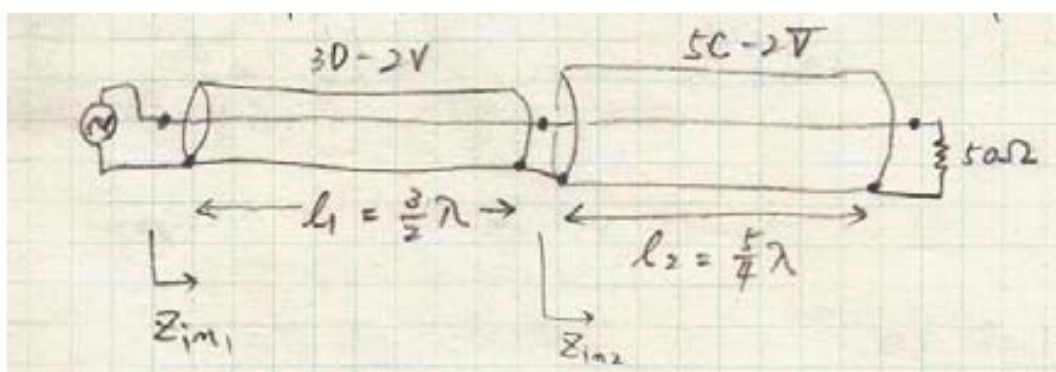


図 3: 回路図

Python で伝達関数の計算, 周波数特性のグラフを出力するプログラムをソースコード 1 に示す.

ソースコード 1: 周波数特性

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import cmath
4
5
6 def calculateTheta(frequency_Hz, cableInfo):
7     """
8     伝搬定数 と同軸ケーブルの長さ $l$ の積を求める
9      $G = 0$ で考える
10
11     Parameters
12     -----
13     frequency_Hz : float
14         周波数 (Hz)
15     cableInfo : dictionary
16         ケーブルの仕様
17     """
18     omega = 2 * np.pi * frequency_Hz
19
20     R_ohmPerM = cableInfo["resistance"] * 1000 # /m
21     L = 1.31 * 10 ** -7 # H/m
22     C_FperM = cableInfo["capacitance"] * 10 ** -12 # F/m
23     G = 0
24
25     # cmath を使わないとエラーが返ってくる
26     gamma = cmath.sqrt((R_ohmPerM + omega * L * 1j) * (G + omega * C_FperM *
27         1j))
28     theta = gamma * cableInfo["cableLength"]
29     return theta
30
31 def createFMatrixForDcc(cableInfo, theta):
32     """
33     分布定数回路の $F$ 行列を求める
34
35     Parameters
36     -----
37     cableInfo : dictionary
38         ケーブルの仕様
39     theta : float
40         伝搬定数 と同軸ケーブルの長さ $l$ の積
41     """
42     return np.array(
43         [
44             [cmath.cosh(theta), cableInfo["impedance"] * cmath.sinh(theta)],
45             [cmath.sinh(theta) / cableInfo["impedance"], cmath.cosh(theta)],
46         ]
47     )
48
49
50 def calculateInputImpedanceByFMatrix(Zr, frequency, cableInfo):
51     """
52     受電端に抵抗を接続した分布定数回路の入力インピーダンスを求める
53     与えられた周波数から入力インピーダンスを求める
54
55     Parameters
```

```

56  -----
57  Zr : float
58      受電端のインピーダンス
59  frequency : float
60      周波数
61  cableInfo : dictionary
62      ケーブルの仕様
63  """
64  # lを求める
65  theta = calculateTheta(frequency, cableInfo)
66  # 分布定数回路のF行列
67  f_matrix_dcc = createFMatrixForDcc(cableInfo, theta)
68  # 受電端のZrのF行列
69  f_matrix_Zr = np.array(
70      [
71          [1, 0],
72          [1 / Zr, 1],
73      ]
74  )
75
76  # 受信端にZrを接続した場合のf行列
77  f_matrix = np.dot(f_matrix_dcc, f_matrix_Zr)
78
79  return abs(f_matrix[0, 0] / f_matrix[1, 0])
80
81
82  def createTransferFunction(Zr, frequency, cableInfo):
83      """
84      受電端に抵抗を接続した分布定数回路の伝達関数を求める
85
86      Parameters
87      -----
88      Zr : float
89          受電端のインピーダンス
90      frequency : float
91          周波数
92      cableInfo : dictionary
93          ケーブルの仕様
94      """
95      theta = calculateTheta(frequency, cableInfo)
96
97      R1 = 0 # 入力側の抵抗は0で考える
98      R2 = Zr
99
100     f_matrix_dcc = createFMatrixForDcc(cableInfo, theta)
101     A = f_matrix_dcc[0][0]
102     B = f_matrix_dcc[0][1]
103     C = f_matrix_dcc[1][0]
104     D = f_matrix_dcc[1][1]
105
106     return 1 / (A + B / R2 + R1 * C + (R1 / R2) * D)
107
108
109  # 5C-2V
110  alpha2_1mhz = 7.6
111  alpha2_10mhz = 25
112  alpha2_200mhz = 125

```

```

113 cable_5c2v = {
114     "cableLength": 5 / 4,
115     "impedance": 75, # 同軸ケーブルのインピーダンス
116     "capacitance": 67, # (nF/km)
117     "resistance": 35.9, # (M /km?)
118     "alphas": [alpha2_1mhz, alpha2_10mhz, alpha2_200mhz],
119 }
120
121 # 3D-2V
122 alpha1_1mhz = 13
123 alpha1_10mhz = 44
124 alpha1_200mhz = 220
125 cable_3d2v = {
126     "cableLength": 3 / 2,
127     "impedance": 50, # 同軸ケーブルのインピーダンス
128     "capacitance": 100, # (nF/km)
129     "resistance": 33.3, # (M /km?) https://www.systemgear.jp/kantsu/3c2v.php
130     "alphas": [alpha1_1mhz, alpha1_10mhz, alpha1_200mhz],
131 }
132
133 # 受端のインピーダンス
134 Zr = 50
135
136 # 単位はMHz (= 1 x 10^6 Hz)
137 frequencies_Hz = range(1, 4 * 10 ** 5, 100)
138 frequencies = frequencies_Hz
139
140 transferFunctions1 = []
141 # 周波数ごとに伝達関数を求める
142 for frequency in frequencies:
143     # 5C-2V + Zr の回路の入カインピーダンスを求める
144     inputImpedance2 = calculateInputImpedanceByFMatrix(
145         Zr,
146         frequency,
147         cable_5c2v,
148     )
149
150     # 回路全体の伝達関数を求める
151     # transferFunctions2 = createTransferFunction(Zr, Z02, l2, frequency, alphas2)
152
153     # 5C-2V + Zr の回路の入カインピーダンスを受電端側の抵抗 Zr とする
154     transferFunction1 = createTransferFunction(inputImpedance2, frequency,
155         cable_3d2v)
156     transferFunctions1.append(transferFunction1)
157
158 # 周波数特性
159 fig, ax = plt.subplots()
160 ax.plot(frequencies, list(map(abs, transferFunctions1)), label="Gain")
161 ax.set_xlabel("frequency [Hz]")
162 ax.set_ylabel("Gain")
163 plt.show()

```

ソースコード 1 によって得られた周波数特性を図 4 に示す。

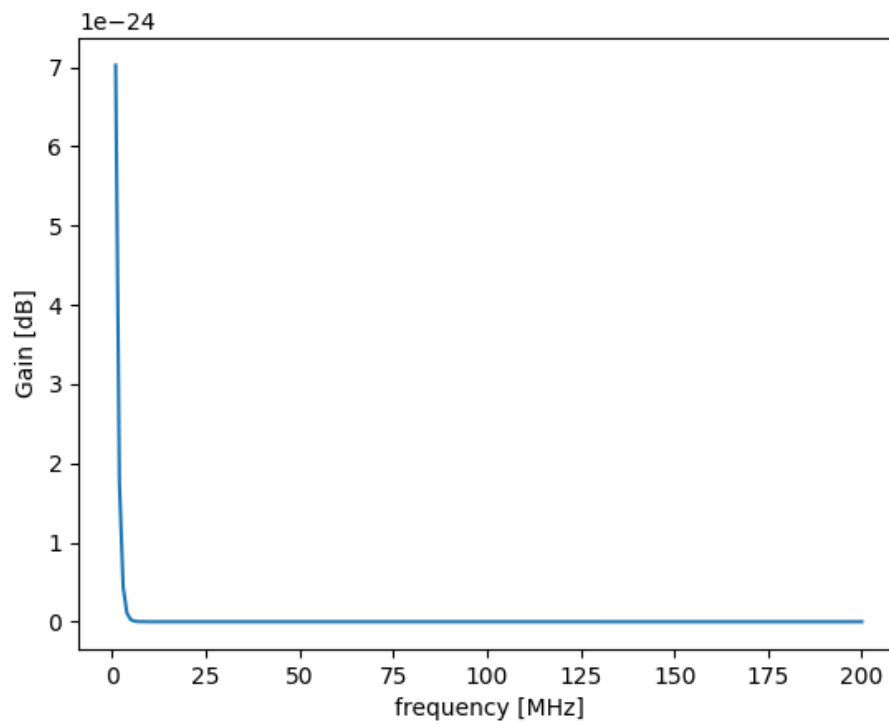


図 4: 周波数特性 (1MHz ~200MHz)

図 4 より, 周波数が 10MHz を超えた辺りからゲインが 0 になっているのでプロットする周波数の値域を 1MHz ~10MHz に変更した物を図 5 に示す.

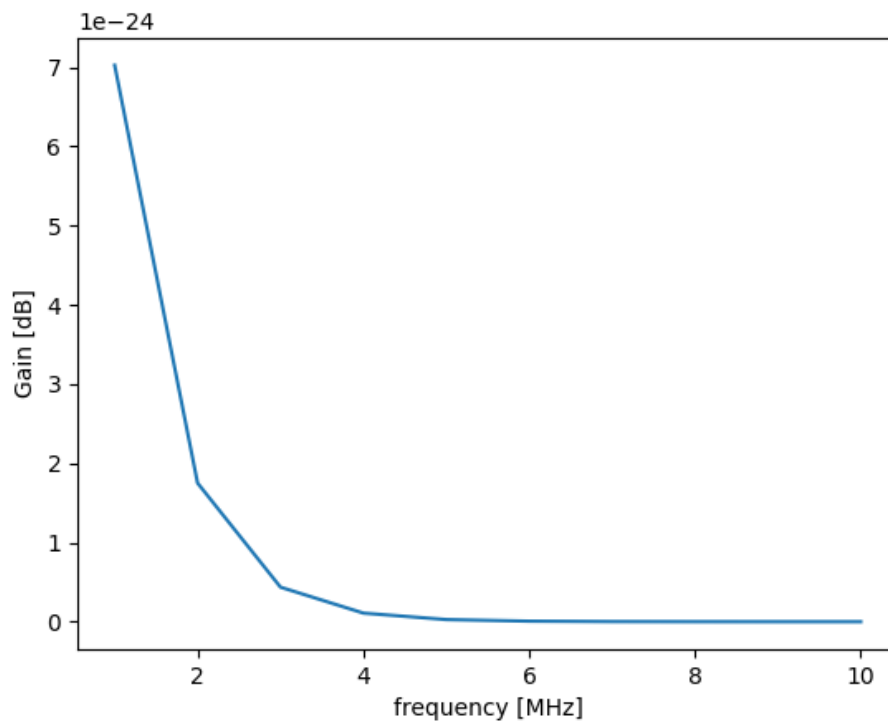


図 5: 周波数特性 (1MHz ~10MHz)

3 おわりに

今回は分布定数線路の周波数特性を調べたが, ローパスフィルターの周波数特性のようにはなかった.

参考文献

[1] 都築, "2020Q4-応用通信工学 II-都築 ", moodle 内, 参照 October 20,2021.