



**INSTITUTO
FEDERAL**

Santa Catarina

Câmpus
São José

Calculo de sobreposição de espectro

Redes de Transmissão

Arthur Cadore Matuella Barcella

25 de Agosto de 2024

Engenharia de Telecomunicações - IFSC-SJ

Sumário

1. Introdução:	3
2. Frequências utilizadas:	3
2.1. Comprimentos de entrada:	3
2.2. Conversão para frequência:	3
3. Alterando o script python:	3
3.1. Script original:	4
3.2. Adição do calculo de sobreposição:	4
3.3. Código final:	5
4. Resultados:	6
4.1. Tolerância de 0.000435 THz	6
4.1.1. Arranjos:	6
4.1.2. Sobreposições:	7
4.1.3. Resultados:	7
4.2. Tolerância de 0.00435 THz	7
4.2.1. Arranjos:	7
4.2.2. Sobreposições:	8
4.2.3. Resultado:	9
5. Conclusão:	9

1. Introdução:

O objetivo deste relatório é realizar o cálculo de sobreposição de espectro utilizando o script python fornecido pelo professor, verificando quais canais possuem sobreposição e quais são as frequências que estão próximas.

2. Frequências utilizadas:

Para o calculo de sobreposição, é necessário primeiramente obter as frequências de cada um dos canais. Para isso, é necessário converter os comprimentos de onda fornecidos em frequência.

2.1. Comprimentos de entrada:

Os comprimentos de onda fornecidos para verificação são os seguintes:

$$\lambda_1 - 1528,77 \text{ nm} \quad (1)$$

$$\lambda_2 - 1529,55 \text{ nm} \quad (2)$$

$$\lambda_3 - 1530,33 \text{ nm} \quad (3)$$

$$\lambda_4 - 1531,12 \text{ nm} \quad (4)$$

2.2. Conversão para frequência:

Considerando a velocidade da luz como 299.792.458 m/s, temos como que a frequência de cada uma das ondas é dada por:

$$F_1 = \frac{299792458}{1528.77 * 10^{-9}} = 196100432373738,3 \text{ ou } 196,1004323737383 \text{ THz} \quad (5)$$

$$F_2 = \frac{299792458}{1529.55 * 10^{-9}} = 196000430191886,5 \text{ ou } 196,0004301918865 \text{ THz} \quad (6)$$

$$F_3 = \frac{299792458}{1530.33 * 10^{-9}} = 195900529951056,3 \text{ ou } 195,9005299510563 \text{ THz} \quad (7)$$

$$F_4 = \frac{299792458}{1531.12 * 10^{-9}} = 195799452688228,2 \text{ ou } 195,7994526882282 \text{ THz} \quad (8)$$

3. Alterando o script python:

Para verificar a sobreposição de espectro, é necessário utilizar um script para verificar se a frequência calculada está próxima de alguma das frequências originais.

3.1. Script original:

Foi fornecido um script base como referência passado pelo professor para verificação de sobreposição de espectro. O mesmo está apresentado abaixo:

```
1 # IFSC Câmpus São José
2 # Engenharia de Telecomunicações
3 # RTR029007 Redes de Transmissão
4 # Professor: Fábio Alexandre de Souza
5 # Four Wave Mixing
6
7 import os
8 import itertools
9
10 print('=====Início=====')
11 print('Ff = fi + fj - fk, onde i dif k e j dif k')
12
13 n = int(input('Digite n: '))
14
15 canais = list()
16
17 for cont in range(0,n):
18     canais.append(float(input('Digite f em THz: ')))
19 c = 0;
20
21 print('=====Arranjos=====')
22
23 for f in itertools.product(range(0, n), repeat=3):
24     if f[0] != f[2] and f[1] != f[2]:
25         c = c+1
26         print('\nArranjo ',c, '=', '[' ,canais[f[0]], ' ',
27             canais[f[1]], ' - ', canais[f[2]], ' ]')
28         Ff = canais[f[0]] + canais[f[1]] - canais[f[2]]
29         print('Ff = ', round(Ff,2))
30         if Ff in canais:
31             print("Sobreposição!")
32
33 numar = c
34 print('número de conjuntos = ',numar)
35
36 print('=====')
37 print('=====FIM=====')
```

3.2. Adição do calculo de sobreposição:

No script, foi adicionado uma função para verificar a sobreposição de espectro, que verifica se a frequência calculada está próxima de alguma das frequências originais:

```
1 # Autor: Arthur Cadore M. Barcella
2 # Função para verificar sobreposição com tolerância
3 def verificar_sobreposicao(canais, resultados, tolerancia=0.000435):
4     sobreposicoes = []
5     canais_set = set(canais)
6     for f, arranjo in resultados:
7         for canal in canais:
8             if abs(f - canal) <= tolerancia:
```

```

9             sobreposicoes.append((arranjo, f, canal))
10            break
11    return sobreposicoes

```

3.3. Código final:

Uma vez editado, o código final para verificação de sobreposição de espectro é o seguinte, além de adicionar a função de verificação de sobreposição, outras alterações foram realizadas:

```

1  import os
2  import itertools
3
4  n = 4
5  canais = [
6      196.1004323737383,
7      196.0004301918865,
8      195.9005299510563,
9      195.7994526882282
10 ]
11
12 def verificar_sobreposicao(canais, resultados, tolerancia=0.000435):
13     sobreposicoes = []
14     canais_set = set(canais)
15     for f, arranjo in resultados:
16         for canal in canais:
17             if abs(f - canal) <= tolerancia:
18                 sobreposicoes.append((arranjo, f, canal))
19                 break
20     return sobreposicoes
21
22 resultados = []
23 c = 0
24
25 print('=====Arranjos=====')
26 for f in itertools.product(range(0, n), repeat=3):
27     if f[0] != f[2] and f[1] != f[2]:
28         c += 1
29         arranjo = [canais[f[0]], canais[f[1]], canais[f[2]]]
30         Ff = canais[f[0]] + canais[f[1]] - canais[f[2]]
31         resultados.append((round(Ff, 2), arranjo))
32         print('\nArranjo ', c, '=', '[' + arranjo[0], arranjo[1], arranjo[2],
33             ']\n')
34         print('Ff = ', round(Ff, 2))
35
36 sobreposicoes = verificar_sobreposicao(canais, resultados)
37 num_sobreposicoes = len(sobreposicoes)
38
39 print('=====Sobreposições=====')
40 if sobreposicoes:
41     print('\nCanais com Sobreposição:')
42     for arranjo, freq_calculada, freq_original in sobreposicoes:
43         print(f'Arranjo {arranjo} tem sobreposição com Ff = {freq_calculada}
44             THz (próximo de {freq_original} THz)')
45
46 print('=====Resultados=====')

```

```

45 print('Número de conjuntos = ', c)
46 print('Número de sobreposições = ', num_sobreposicoes)

```

4. Resultados:

Uma vez com o script finalizado, é possível realizar a execução do mesmo para verificar a sobreposição de espectro. Para isso, devemos considerar um valor de tolerância a ser utilizado para decidir se o canal está ou não sobrepondo um outro canal.

4.1. Tolerância de 0.000435 THz

Utilizando a tolerância de 0.000435 THz, temos os seguintes resultados:

4.1.1. Arranjos:

Foram obtidos os seguintes arranjos:

```

1  Arranjo 1 = [ 196.100432373 196.100432373 196.000430191 ] -> Ff = 196.2
2  Arranjo 2 = [ 196.100432373 196.100432373 195.900529951 ] -> Ff = 196.3
3  Arranjo 3 = [ 196.100432373 196.100432373 195.799452688 ] -> Ff = 196.4
4  Arranjo 4 = [ 196.100432373 196.000430191 195.900529951 ] -> Ff = 196.2
5  Arranjo 5 = [ 196.100432373 196.000430191 195.799452688 ] -> Ff = 196.3
6  Arranjo 6 = [ 196.100432373 195.900529951 196.000430191 ] -> Ff = 196.0
7  Arranjo 7 = [ 196.100432373 195.900529951 195.799452688 ] -> Ff = 196.2
8  Arranjo 8 = [ 196.100432373 195.799452688 196.000430191 ] -> Ff = 195.9
9  Arranjo 9 = [ 196.100432373 195.799452688 195.900529951 ] -> Ff = 196.0
10 Arranjo 10 = [ 196.000430191 196.100432373 195.900529951 ] -> Ff = 196.2
11 Arranjo 11 = [ 196.000430191 196.100432373 195.799452688 ] -> Ff = 196.3
12 Arranjo 12 = [ 196.000430191 196.000430191 196.100432373 ] -> Ff = 195.9
13 Arranjo 13 = [ 196.000430191 196.000430191 195.900529951 ] -> Ff = 196.1
14 Arranjo 14 = [ 196.000430191 196.000430191 195.799452688 ] -> Ff = 196.2
15 Arranjo 15 = [ 196.000430191 195.900529951 196.100432373 ] -> Ff = 195.8
16 Arranjo 16 = [ 196.000430191 195.900529951 195.799452688 ] -> Ff = 196.1
17 Arranjo 17 = [ 196.000430191 195.799452688 196.100432373 ] -> Ff = 195.7
18 Arranjo 18 = [ 196.000430191 195.799452688 195.900529951 ] -> Ff = 195.9
19 Arranjo 19 = [ 195.900529951 196.100432373 196.000430191 ] -> Ff = 196.0
20 Arranjo 20 = [ 195.900529951 196.100432373 195.799452688 ] -> Ff = 196.2
21 Arranjo 21 = [ 195.900529951 196.000430191 196.100432373 ] -> Ff = 195.8
22 Arranjo 22 = [ 195.900529951 196.000430191 195.799452688 ] -> Ff = 196.1
23 Arranjo 23 = [ 195.900529951 195.900529951 196.100432373 ] -> Ff = 195.7
24 Arranjo 24 = [ 195.900529951 195.900529951 196.000430191 ] -> Ff = 195.8
25 Arranjo 25 = [ 195.900529951 195.900529951 195.799452688 ] -> Ff = 196.0
26 Arranjo 26 = [ 195.900529951 195.799452688 196.100432373 ] -> Ff = 195.6
27 Arranjo 27 = [ 195.900529951 195.799452688 196.000430191 ] -> Ff = 195.7
28 Arranjo 28 = [ 195.799452688 196.100432373 196.000430191 ] -> Ff = 195.9
29 Arranjo 29 = [ 195.799452688 196.100432373 195.900529951 ] -> Ff = 196.0
30 Arranjo 30 = [ 195.799452688 196.000430191 196.100432373 ] -> Ff = 195.7
31 Arranjo 31 = [ 195.799452688 196.000430191 195.900529951 ] -> Ff = 195.9
32 Arranjo 32 = [ 195.799452688 195.900529951 196.100432373 ] -> Ff = 195.6
33 Arranjo 33 = [ 195.799452688 195.900529951 196.000430191 ] -> Ff = 195.7
34 Arranjo 34 = [ 195.799452688 195.799452688 196.100432373 ] -> Ff = 195.5
35 Arranjo 35 = [ 195.799452688 195.799452688 196.000430191 ] -> Ff = 195.6

```

```
36 Arranjo 36 = [ 195.799452688 195.799452688 195.900529951 ] -> Ff = 195.7
```

4.1.2. Sobreposições:

A partir dos arranjos obtidos, foram verificadas as sobreposições com as frequências de entrada, que são as seguintes:

```
1 Arranjo [196.100432373, 195.900529951, 196.000430191] -> Sobreposição com
  Ff = 196.0 (Próximo de: 196.000430191 THz)
2 Arranjo [196.100432373, 195.799452688, 195.900529951] -> Sobreposição com
  Ff = 196.0 (Próximo de: 196.000430191 THz)
3 Arranjo [196.000430191, 196.000430191, 195.900529951] -> Sobreposição com
  Ff = 196.1 (Próximo de: 196.100432373 THz)
4 Arranjo [196.000430191, 195.900529951, 195.799452688] -> Sobreposição com
  Ff = 196.1 (Próximo de: 196.100432373 THz)
5 Arranjo [195.900529951, 196.100432373, 196.000430191] -> Sobreposição com
  Ff = 196.0 (Próximo de: 196.000430191 THz)
6 Arranjo [195.900529951, 196.000430191, 195.799452688] -> Sobreposição com
  Ff = 196.1 (Próximo de: 196.100432373 THz)
7 Arranjo [195.900529951, 195.900529951, 195.799452688] -> Sobreposição com
  Ff = 196.0 (Próximo de: 196.000430191 THz)
8 Arranjo [195.799452688, 196.100432373, 195.900529951] -> Sobreposição com
  Ff = 196.0 (Próximo de: 196.000430191 THz)
```

4.1.3. Resultados:

Com os resultados obtidos, temos que:

```
1 Número de conjuntos = 36
2 Número de sobreposições = 8
```

4.2. Tolerância de 0.00435 THz

Utilizando a tolerância de 0.00435 THz, temos os seguintes resultados:

4.2.1. Arranjos:

Foram obtidos os seguintes arranjos:

```
1 Arranjo 1 = [ 196.100432373 196.100432373 196.000430191 ] -> Ff = 196.2
2 Arranjo 2 = [ 196.100432373 196.100432373 195.900529951 ] -> Ff = 196.3
3 Arranjo 3 = [ 196.100432373 196.100432373 195.799452688 ] -> Ff = 196.4
4 Arranjo 4 = [ 196.100432373 196.000430191 195.900529951 ] -> Ff = 196.2
5 Arranjo 5 = [ 196.100432373 196.000430191 195.799452688 ] -> Ff = 196.3
6 Arranjo 6 = [ 196.100432373 195.900529951 196.000430191 ] -> Ff = 196.0
7 Arranjo 7 = [ 196.100432373 195.900529951 195.799452688 ] -> Ff = 196.2
8 Arranjo 8 = [ 196.100432373 195.799452688 196.000430191 ] -> Ff = 195.9
9 Arranjo 9 = [ 196.100432373 195.799452688 195.900529951 ] -> Ff = 196.0
10 Arranjo 10 = [ 196.000430191 196.100432373 195.900529951 ] -> Ff = 196.2
```

```

11 Arranjo 11 = [ 196.000430191 196.100432373 195.799452688 ] -> Ff = 196.3
12 Arranjo 12 = [ 196.000430191 196.000430191 196.100432373 ] -> Ff = 195.9
13 Arranjo 13 = [ 196.000430191 196.000430191 195.900529951 ] -> Ff = 196.1
14 Arranjo 14 = [ 196.000430191 196.000430191 195.799452688 ] -> Ff = 196.2
15 Arranjo 15 = [ 196.000430191 195.900529951 196.100432373 ] -> Ff = 195.8
16 Arranjo 16 = [ 196.000430191 195.900529951 195.799452688 ] -> Ff = 196.1
17 Arranjo 17 = [ 196.000430191 195.799452688 196.100432373 ] -> Ff = 195.7
18 Arranjo 18 = [ 196.000430191 195.799452688 195.900529951 ] -> Ff = 195.9
19 Arranjo 19 = [ 195.900529951 196.100432373 196.000430191 ] -> Ff = 196.0
20 Arranjo 20 = [ 195.900529951 196.100432373 195.799452688 ] -> Ff = 196.2
21 Arranjo 21 = [ 195.900529951 196.000430191 196.100432373 ] -> Ff = 195.8
22 Arranjo 22 = [ 195.900529951 196.000430191 195.799452688 ] -> Ff = 196.1
23 Arranjo 23 = [ 195.900529951 195.900529951 196.100432373 ] -> Ff = 195.7
24 Arranjo 24 = [ 195.900529951 195.900529951 196.000430191 ] -> Ff = 195.8
25 Arranjo 25 = [ 195.900529951 195.900529951 195.799452688 ] -> Ff = 196.0
26 Arranjo 26 = [ 195.900529951 195.799452688 196.100432373 ] -> Ff = 195.6
27 Arranjo 27 = [ 195.900529951 195.799452688 196.000430191 ] -> Ff = 195.7
28 Arranjo 28 = [ 195.799452688 196.100432373 196.000430191 ] -> Ff = 195.9
29 Arranjo 29 = [ 195.799452688 196.100432373 195.900529951 ] -> Ff = 196.0
30 Arranjo 30 = [ 195.799452688 196.000430191 196.100432373 ] -> Ff = 195.7
31 Arranjo 31 = [ 195.799452688 196.000430191 195.900529951 ] -> Ff = 195.9
32 Arranjo 32 = [ 195.799452688 195.900529951 196.100432373 ] -> Ff = 195.6
33 Arranjo 33 = [ 195.799452688 195.900529951 196.000430191 ] -> Ff = 195.7
34 Arranjo 34 = [ 195.799452688 195.799452688 196.100432373 ] -> Ff = 195.5
35 Arranjo 35 = [ 195.799452688 195.799452688 196.000430191 ] -> Ff = 195.6
36 Arranjo 36 = [ 195.799452688 195.799452688 195.900529951 ] -> Ff = 195.7

```

4.2.2. Sobreposições:

A partir dos arranjos obtidos, foram verificadas as sobreposições com as frequências de entrada, que são as seguintes:

```

1 Canais com Sobreposição:
2 Arranjo [196.100432373, 195.900529951, 196.000430191] -> Sobreposição com
  Ff = 196.0 (próximo de: 196.000430191 THz)
3 Arranjo [196.100432373, 195.799452688, 196.000430191] -> Sobreposição com
  Ff = 195.9 (próximo de: 195.900529951 THz)
4 Arranjo [196.100432373, 195.799452688, 195.900529951] -> Sobreposição com
  Ff = 196.0 (próximo de: 196.000430191 THz)
5 Arranjo [196.000430191, 196.000430191, 196.100432373] -> Sobreposição com
  Ff = 195.9 (próximo de: 195.900529951 THz)
6 Arranjo [196.000430191, 196.000430191, 195.900529951] -> Sobreposição com
  Ff = 196.1 (próximo de: 196.100432373 THz)
7 Arranjo [196.000430191, 195.900529951, 196.100432373] -> Sobreposição com
  Ff = 195.8 (próximo de: 195.799452688 THz)
8 Arranjo [196.000430191, 195.900529951, 195.799452688] -> Sobreposição com
  Ff = 196.1 (próximo de: 196.100432373 THz)
9 Arranjo [196.000430191, 195.799452688, 195.900529951] -> Sobreposição com
  Ff = 195.9 (próximo de: 195.900529951 THz)
10 Arranjo [195.900529951, 196.100432373, 196.000430191] -> Sobreposição com
  Ff = 196.0 (próximo de: 196.000430191 THz)
11 Arranjo [195.900529951, 196.000430191, 196.100432373] -> Sobreposição com
  Ff = 195.8 (próximo de: 195.799452688 THz)

```



```

12 Arranjo [195.900529951, 196.000430191, 195.799452688] -> Sobreposição com
    Ff = 196.1 (próximo de: 196.100432373 THz)
13 Arranjo [195.900529951, 195.900529951, 196.000430191] -> Sobreposição com
    Ff = 195.8 (próximo de: 195.799452688 THz)
14 Arranjo [195.900529951, 195.900529951, 195.799452688] -> Sobreposição com
    Ff = 196.0 (próximo de: 196.000430191 THz)
15 Arranjo [195.799452688, 196.100432373, 196.000430191] -> Sobreposição com
    Ff = 195.9 (próximo de: 195.900529951 THz)
16 Arranjo [195.799452688, 196.100432373, 195.900529951] -> Sobreposição com
    Ff = 196.0 (próximo de: 196.000430191 THz)
17 Arranjo [195.799452688, 196.000430191, 195.900529951] -> Sobreposição com
    Ff = 195.9 (próximo de: 195.900529951 THz)

```

4.2.3. Resultado:

Com os resultados obtidos, temos que:

```

1 Número de conjuntos = 36
2 Número de sobreposições = 16

```

5. Conclusão:

A partir dos conceitos vistos, alterações de código, e resultados obtidos, podemos concluir que a verificação de sobreposição de espectro é uma ferramenta importante para a análise de redes de transmissão, permitindo identificar quais canais estão próximos e podem causar interferência entre si. A partir dos resultados obtidos, é possível verificar que a tolerância utilizada influencia diretamente na quantidade de sobreposições encontradas, sendo necessário ajustar o valor de tolerância de acordo com a aplicação desejada.

Quanto maior a tolerância utilizada, maior a quantidade de sobreposições encontradas, sendo necessário um ajuste fino para encontrar o equilíbrio entre a quantidade de sobreposições e a precisão desejada, de acordo com o tamanho do canal sendo utilizado na transmissão.