

Atividade 4

Leonardo Santos - GRR20196154

Primeiramente foi realizada a normalização dos dados, utilizando esse trecho de código ilustrado pela Figura 1 a seguir:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.io import loadmat
4
5
6 # Carregar arquivo .mat
7 mat = loadmat('in_out_SMR2_direta.mat')
8
9 in_data_ext = mat['in_extraction']
10 out_data_ext = mat['out_extraction']
11 in_data_val = mat['in_validation']
12 out_data_val = mat['out_validation']
13
14 max_value = np.concatenate((max(abs(in_data_ext)), max(abs(out_data_ext)), max(abs(in_data_val)), max(abs(out_data_val))))
15
16
17 in_ext = in_data_ext/max_value
18 out_ext = out_data_ext/max_value
19 in_val = in_data_val/max_value
20 out_val = out_data_val/max_value
21
```

Figura 1: Código de normalização dos dados

Em seguida foi utilizado o trecho de código para avaliar se os valores se encontravam entre os valores de -1 e 1. O resultado está ilustrado pela Figura 2 a seguir:

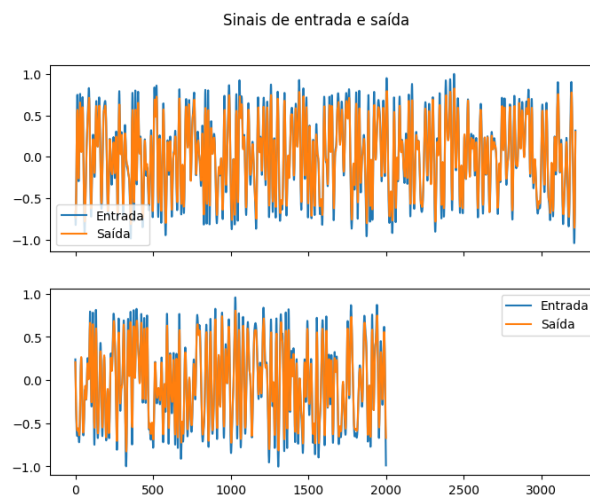


Figura 2: Gráficos entrada e saída

Com os dados normalizados foi realizada a extração dos coeficientes, utilizando o seguinte trecho de código ilustrado pela Figura 3 a seguir:

```
1 P = 4
2 M = 2
3
4 XX_ext_norm = mp(P, M, in_ext_norm)
5 coefficients, _ , _ = np.linalg.lstsq(XX_ext_norm, out_ext_norm[M:], rcond=None)
6
```

Figura 3: Cálculo das matrizes de Coeficientes

Em seguida foi realizada a extração da matriz XX de validação, com a virgula fixa utilizando o seguinte trecho de código ilustrado pela Figura 4.

```

1
2 def mp_int(P, M, xn, bits):
3     L = xn.shape
4     XX = np.zeros((L[0] - M, P * (M+1)), dtype=np.complex128)
5     for l in range(M+1, L[0]):
6         for p in range(1, P+1):
7             for m in range(0, M+1):
8                 A = np.real(xn[l-m])[0]
9                 B = np.imag(xn[l-m])[0]
10                modulo_power = 2**bits
11                modulo_square = readeq(A ** 2 + B ** 2, bits)
12                for _ in range(1, p):
13                    modulo_power = readeq(modulo_power * modulo_square, bits)
14                real_part = readeq(A * modulo_power, bits)
15                imag_part = readeq(B * modulo_power, bits)
16                XX[l-M-1, ((p-1)*(M+1))+m] = complex(real_part, imag_part)
17     return XX
18

```

Figura 4: Função que realiza o calculo da matriz XX inteira

e tambem foram desenvolvido a função que realiza a multiplicação de matrizes de numeros inteiros imaginarios conforme ilustrado Figura 5 a seguir:

```

1
2 def MultiplicadorMatrizes(coefficients, XX, precision):
3     max_value = 0
4     coefficients = np.round(coefficients *(2**precision) )
5     result = np.zeros_like(XX, dtype=np.complex128)
6     for i in range(XX.shape[0]):
7         for j in range(XX.shape[1]):
8
9             A = np.real(coefficients[j,0])
10            B = np.imag(coefficients[j,0])
11            C = np.real(XX[i, j])
12            D = np.imag(XX[i, j])
13
14            readequated_real = readeq(A * C - B * D, precision)
15            readequated_imag = readeq(A * D + B * C, precision)
16
17            max_value = max(max_value, readequated_real)
18            max_value = max(max_value, readequated_imag)
19
20            result[i, j] = complex(readequated_real, readequated_imag)
21
22     soma_filas = np.sum(result, axis=1).reshape(-1, 1)
23
24     extra_bits = np.ceil(np.log2(max_value + 1))
25
26     return soma_filas, extra_bits

```

Figura 5: Função que realiza o calculo da matriz XX inteira

Os resultados de MNSE e numeros de bits estão ilustrados pela Figura 6 a seguir:

Bits de precisão: 2,	Bits total: 3,	NMSE: 6.534320285527902 dB
Bits de precisão: 4,	Bits total: 5,	NMSE: -2.4313304098288295 dB
Bits de precisão: 6,	Bits total: 7,	NMSE: -12.881880244656651 dB
Bits de precisão: 8,	Bits total: 9,	NMSE: -22.238818674136134 dB
Bits de precisão: 10,	Bits total: 11,	NMSE: -25.08468599538043 dB
Bits de precisão: 15,	Bits total: 16,	NMSE: -25.31314562589796 dB
Bits de precisão: 20,	Bits total: 21,	NMSE: -25.311210352734655 dB
Bits de precisão: 25,	Bits total: 26,	NMSE: -25.311125235380313 dB

Figura 6: Resultado NMSE

E os graficos resultantes estão disponiveis pelas Figura 7 Figura 8 a seguir:

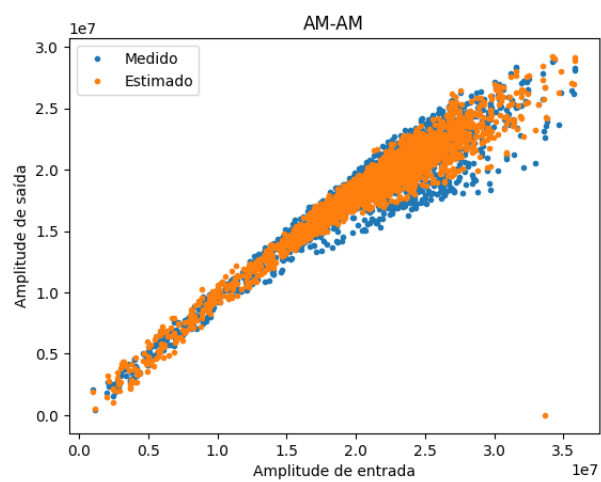


Figura 7: Grafico 1

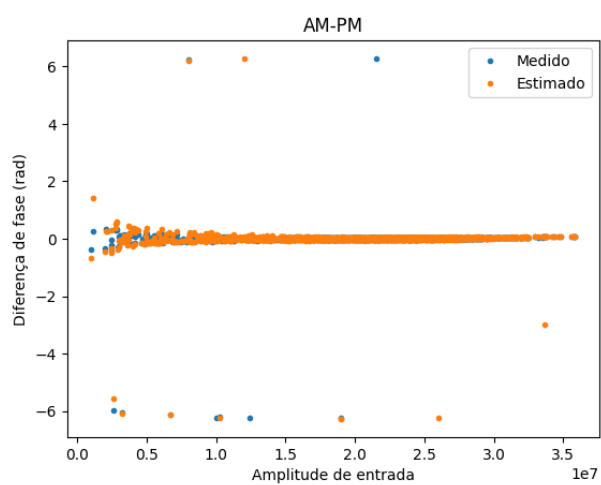


Figura 8: Grafico 2