

МЕТОДИКА ВИМІРУ ДІАГРАМИ СПРЯМОВАНOSTІ МІКРОФОНА**Усик В.В., Тєплоухов Д.Є.*****Національний технічний університет
«Харківський політехнічний інститут», м. Харків***

Сьогодні ринок електроакустичних перетворювачів пропонує величезний вибір сучасних пристроїв, які відповідають будь-якому запиту користувача. Однак, іноді заявлені характеристики та параметри не відповідають дійсності. Не є винятком і такі електроакустичні перетворювачі, як мікрофони, у паспортних даних яких представлені параметри та характеристики, які можуть не відповідати дійсності. Така характеристика, як діаграма спрямованості, дуже важлива для пристрою, оскільки дуже часто визначає область застосування даного мікрофона. Для вимірювання діаграми спрямованості необхідне використання певних акустичних умов та алгоритмів. Сьогодні звичайний користувач не завжди має можливість отримати доступ або створити необхідні умови для проведення експериментів. Таким чином, вимірювання діаграм спрямованості мікрофона вимагає доступніших методик. Необхідні нові методики та сучасні програмні рішення, що скорочують часові та ресурсні витрати на оцінку характеристик електроакустичних перетворювачів.

Запропонована авторами методика може допомогти прискорити та спростити вимірювання в домашніх умовах без використання спеціальних акустичних умов.

Методику можна умовно розділити на два етапи: запис фрагменту звуку, у якому відображено зміну амплітуди гучності в залежності від орієнтації джерела звуку у горизонтальній площині; обробка цього фрагменту у програмі, написаної на Python (рис.1) для побудови діаграми (рис. 2).

```

1 import wave
2 import numpy as np
3 import matplotlib.pyplot as plt
4 from scipy.optimize import curve_fit
5
6 file_path = 'audio.wav'
7 wav_file = wave.open(file_path, 'r')
8
9 num_frames = wav_file.getnframes()
10 frame_rate = wav_file.getframerate()
11 frames = wav_file.readframes(num_frames)
12
13 wav_file.close()
14
15 # Преобразуем данные в массив numpy и нормализуем их
16 signal = np.frombuffer(frames, dtype=np.int16)
17 signal = np.abs(signal)
18 signal = signal.astype(np.float32) - np.min(signal.astype(np.float32))
19
20 # Выберем среднее
21 window_size = 10000
22 num_chunks = len(signal) // window_size
23 signal_filtered = np.zeros(num_chunks)
24
25 for i in range(num_chunks):
26     start = i * window_size
27     end = start + window_size
28     signal_filtered[i] = np.average(signal[start:end])
29
30 ax = plt.subplot(111, projection='polar')
31 ax.set_theta_zero_location("N")
32 ax.set_rscale("log")
33 ax.plot([np.pi/2 / len(signal_filtered) * i for i in range(len(signal_filtered))], signal_filtered)
34 ax.grid(True)
35 plt.title('Waveform')
36 plt.show()
37

```

Рисунок 1 – Фрагмент програмного коду для обробки звукового сигналу

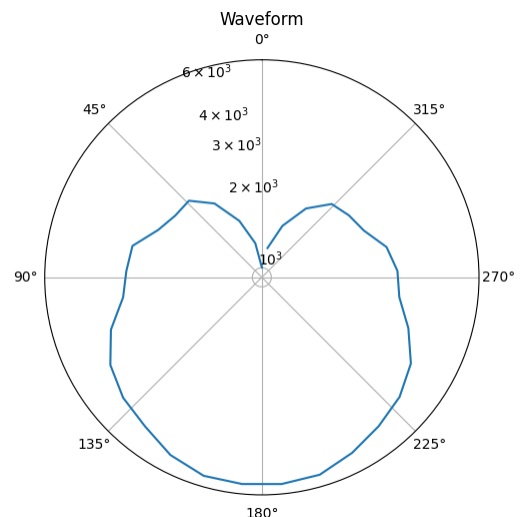


Рисунок 2 – Виміряна діаграма спрямованості мікрофона Shure SM58