|  |
| --- |
| **Experiment 5** |
| **Title:** **Real time audio signal and Spectral analysis**  The music signal has frequency components in the audio frequency range 20 Hz to 20000 Hz of the electromagnetic spectrum. Record or use the recorded music samples of different instruments (at least four) and Write a program to record the music signal and sketch it in time domain, its amplitude spectrum and phase spectrum. Also comment on the results. |
| **Learning Objectives** |
| 1. To understand basic concept of sampling of audio signal 2. To study frequency range of audio signals 3. To study magnitude and phase spectrum of audio signal 4. To have hands on simulation using python language |
| **Prerequisites** |
| 1. Basic understanding of mathematics 2. Basic understanding of signal representation 3. Basics of signal sampling 4. Basic understanding of Python language |
| **Theory** |
| Sampling Theorem:According to Nyquist principle of sampling, in order to remove alising effect signals are sampled with sampling frequency equals to twice of the frequency of signal to be sampled.If F = frequency of signal to be sampledThen, sampling frequency = Fs ≥ 2F |
| **Simulation Code** |
| **import sounddevice as sd**  **from scipy.io.wavfile import read, write**  **import matplotlib.pyplot as plt**  **# Sampling frequency**  **freq = 44000**  **# Recording duration in milliseconds**  **duration = 5**  **# Start recorder with the given values of**  **# duration and sample frequency**  **recording = sd.rec(int(duration \* freq), samplerate=freq, channels=2)**  **# Record audio for the given number of seconds**  **sd.wait()**  **# This will convert the NumPy array to an audio**  **# file with the given sampling frequency**  **write("recording0.wav", freq, recording)**  **Fs, data = read("recording0.wav")**  **data = data[:, 0]**  **print("sampling frequency = ", Fs)**  **plt.subplot(3,1,1)**  **plt.plot(data)**  **plt.title('audio in time domain')**  **plt.xlabel('time t -->')**  **plt.ylabel('magnitude')**  **plt.subplot(3,1,2)**  **plt.magnitude\_spectrum(data, Fs)**  **plt.title('Magnitude spectrum')**  **plt.subplot(3,1,3)**  **plt.phase\_spectrum(data)**  **plt.title('Phase spectrum')**  **plt.show()** |
| **Assignment** |
| **Task1: Write description for code**  **Task2: Vary the parameters of the signals and observe the magnitude and phase spectrum** |
| **Conclusion** |
|  |