INSTITUTE OF AERONAUTICAL ENGINEERING
(Autonomous)
Dundigal, Hyderabad-500 043

Project Based Learning
(Prototype / Design Building)External Evaluation Report

Title of your Idea : **Noise reduction using signal to noise ratio (SNR) reduction**

Thrust Area / Sector : **Communications sector**

Branch : **ECE**

Year / Semester : **3rd Year 5th semester**

Sl.no	Name of the Student	Roll Number	Mobile Number	Signature
1	Saikran Podugu	19951A04D6	+919505595861	

1. Background of the Idea :

Now a days we are facing lots of lots of disturbance with background noise and all the real measurements are disturbed by noise. This includes electronic noise, but can also include An external event that affects the measured phenomenon like – wind, vibrations the Gravitational attraction of the moon, variations of temperature, variations of humidity etc., depending on what is measured and of the sensitivity of the device. It is often possible to reduce the noise by controlling the environment.

Internal electronic noise of measurement systems can be reduced through the use of low noise Amplifiers .When the characteristics of the noise are known and are different from the signal it is possible to use a filter to reduce the noise. A low pass filter (LPF) is an audio signal processor that removes unwanted noise frequencies from a signal above a determined cut off frequency.

All real measurements are disturbed by noise. This includes electronic noise, but can also include external events that affect the measured phenomenon — wind, vibrations, the gravitational attraction of the moon, variations of temperature, variations of humidity, etc., depending on what is measured and of the sensitivity of the device. It is often possible to reduce the noise by controlling the environment.

Internal electronic noise of measurement systems can be reduced through the use of low-noise Amplifiers. When the characteristics of the noise are known and are different from the signal, it is possible to use a filter to reduce the noise.

For example,

A lock in amplifier can extract a narrow bandwidth signal from broadband noise a million times Stronger when the signal is constant or periodic and the noise is random, it is possible to enhance the SNR by averaging the measurements, In this case the noise goes down as the square root of the number of averaged samples.

Here in this project I am using noise reduction algorithm, by taking one sample audio clip which is very clear in audio and into to that clear audio signal I am inserting some multiple noises with different frequency’s and after that reducing the noise by using the SNR reduction method.

This is a very economic technology and can be used in several other fields as well, few are Listed as below:

1. Can be used in measurement in the fields of science and engineering that compares the level of desired to the level of background noise.

2. Can be used in telecommunications for reduction the noise while streaming with others and also useful in reducing the noise which is created in background while making a call or any other interaction.

1. Problem Statement:

In this method the noise can be reduced very easily without losing the original audio frequency and by just reducing the multiple background noises with the use of SNR filter by removing the stationary and non stationary noises from the audio sample without causing any distortion to the original signal.

3. Proposed Solution:

The proposed method deals with Internal electronic noise of measurement systems can be reduced through the use of low noise amplifiers .When the characteristics of the noise are known and are different from the signal it is possible to use a filter to reduce the noise. A low pass filter (LPF) is an audio signal processor that removes unwanted noise frequencies from a signal above a determined cut off frequency and the noise can be reduced very easily without losing the original audio frequency and also by just reducing the multiple background noises with the use of SNR filter by removing the stationary and non stationary noises from the audio sample without causing any distortion to the original signal.

Signal-to-noise ratio is defined as the ratio of the power of a signal (meaningful input) to the power of background noise (meaningless or unwanted input):

$$\mathrm{SNR} = \frac{P_{\mathrm{signal}}}{P_{\mathrm{noise}}}$$

4. Technology concept formulation:

Noise reduction using signal to noise ratio (SNR)

Step 1: Download the required packages from the required sites and I downloaded the required packages from the google collab.

Step 2: This step uses scipy for solving the mathematical problems, noisereduce for reducing the noise from the main source and for inserting audio file I used the soundfile, after that I plotted the audio by using the matplot package library.

Step 3: By loading the data of sample audio and plotting the data by using the matplot for verification I inserted a sample audio noise into it and I verified the signal after insertion of noise and removed the sample noise from the signal and displayed the graph and with audio signal.

5. Prototype of proposed system (UI screens / block diagrams / circuits / designs):

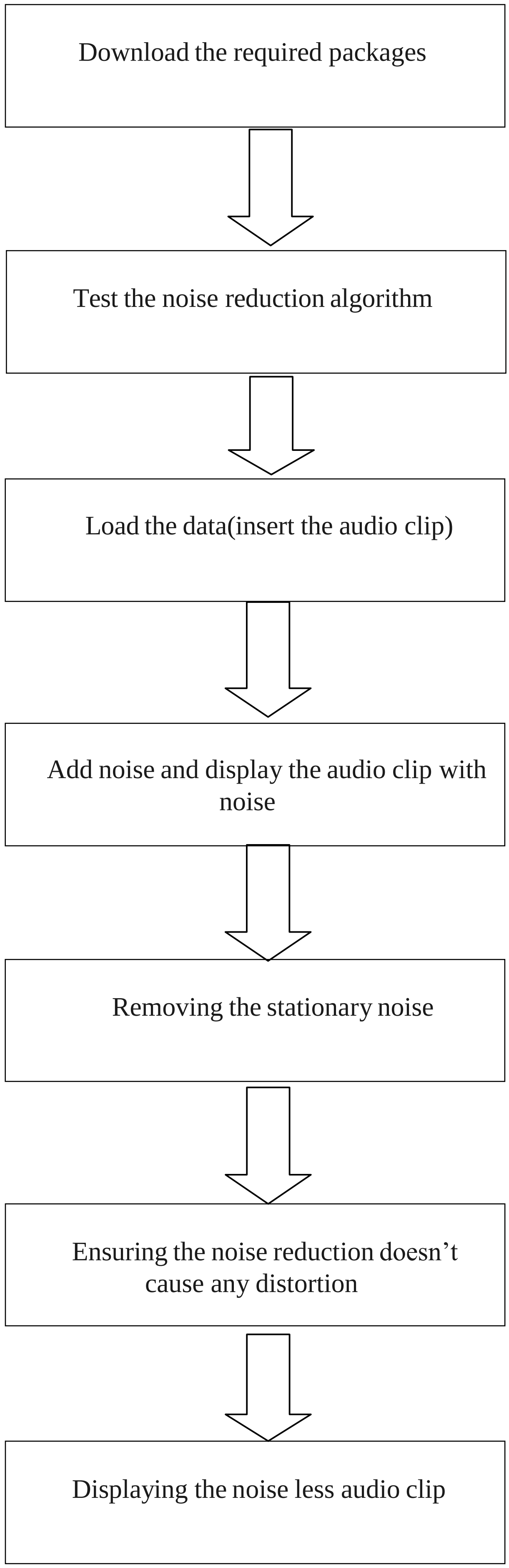
Steps of construction

1. Downloading the required packages.
2. Testing the noise reduction algorithm and view the steps of algorithm.
3. Load the data by inserting the audio clip
4. Add the noise to the audio clip and play the audio clip with noise in background.
5. Remove the stationary noise from the audio clip.
6. Ensuring the noise reduction does not cause distortion.

A MORE DIFFICULT EXAMPLE

- Adding noise to the data
- Plotting the noisy data
- Removing the stationary noise
- Reducing the non stationary noise from the data
- Reduce noise over batches in parallel on long signal
- (optional when the data is long then it is considered)
- Reduce noise on only a subset of long clip (optional)

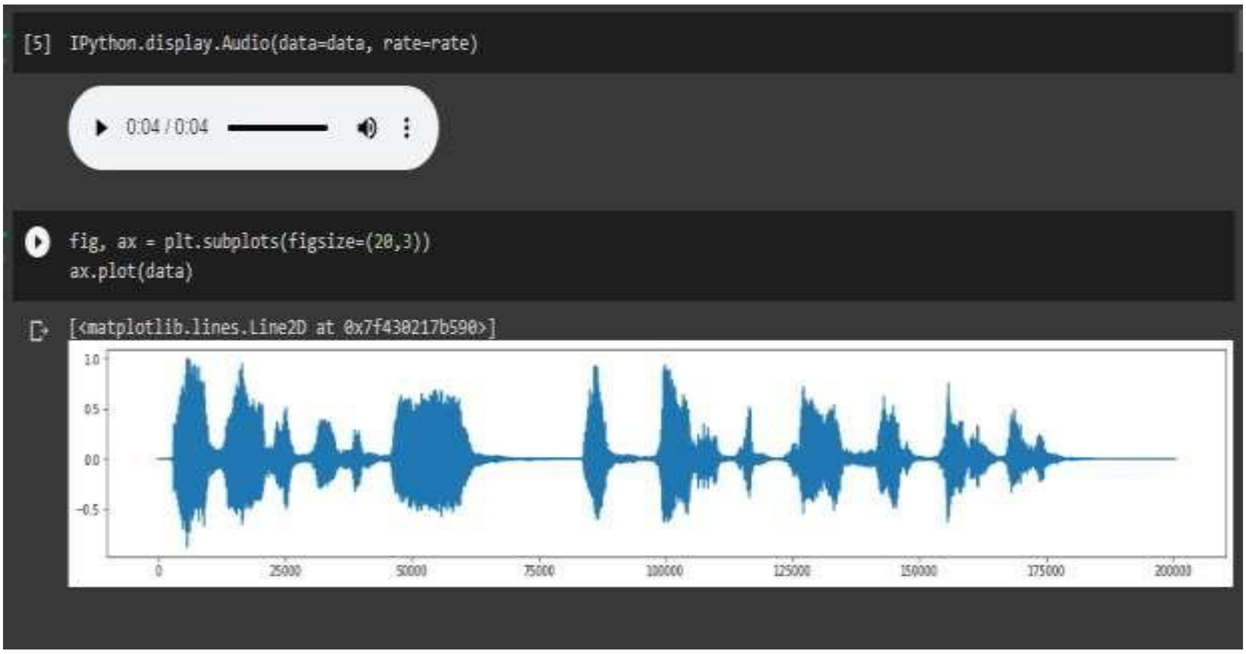
Block diagram behind the working function of this prototype proposed system



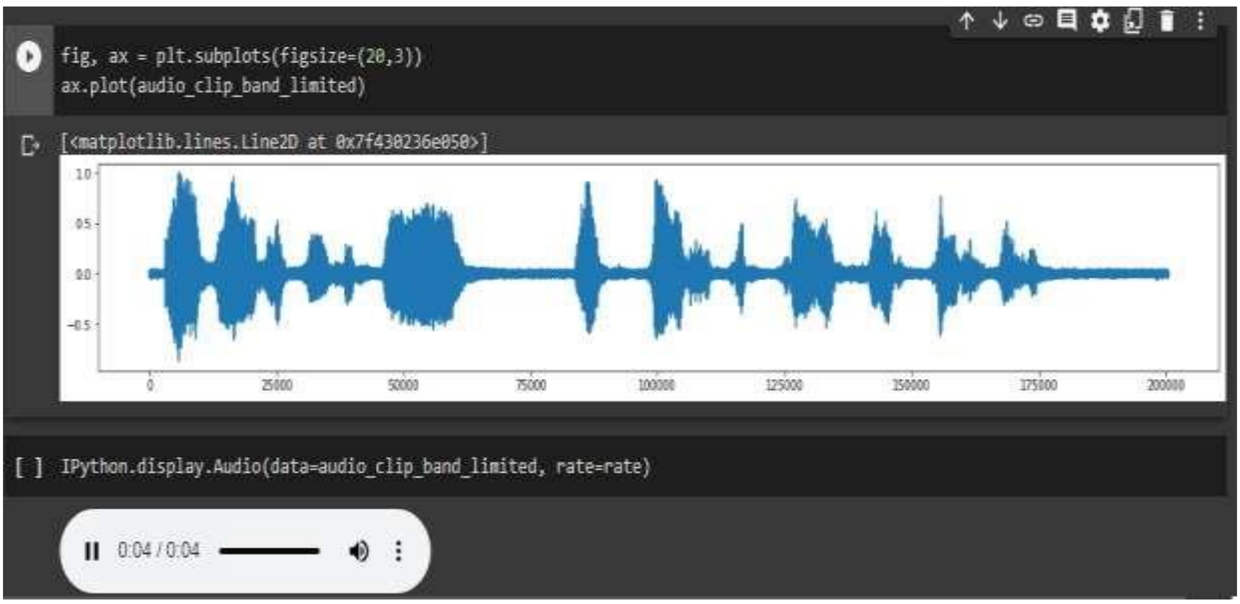
5. Detailed description of prototype / product / project:

Majorly in our day to day life we are facing some problems with noise disturbance that is caused by different environments. I pointed here some problems Civil engineer whenever he is in the construction area he may face some issues while answering the call by background noise, every common face issue with noise while travelling we get some background noise caused by other vehicles or by air to overcome this issue I developed one solution using SNR to reduce noise which is disturbing us from background.

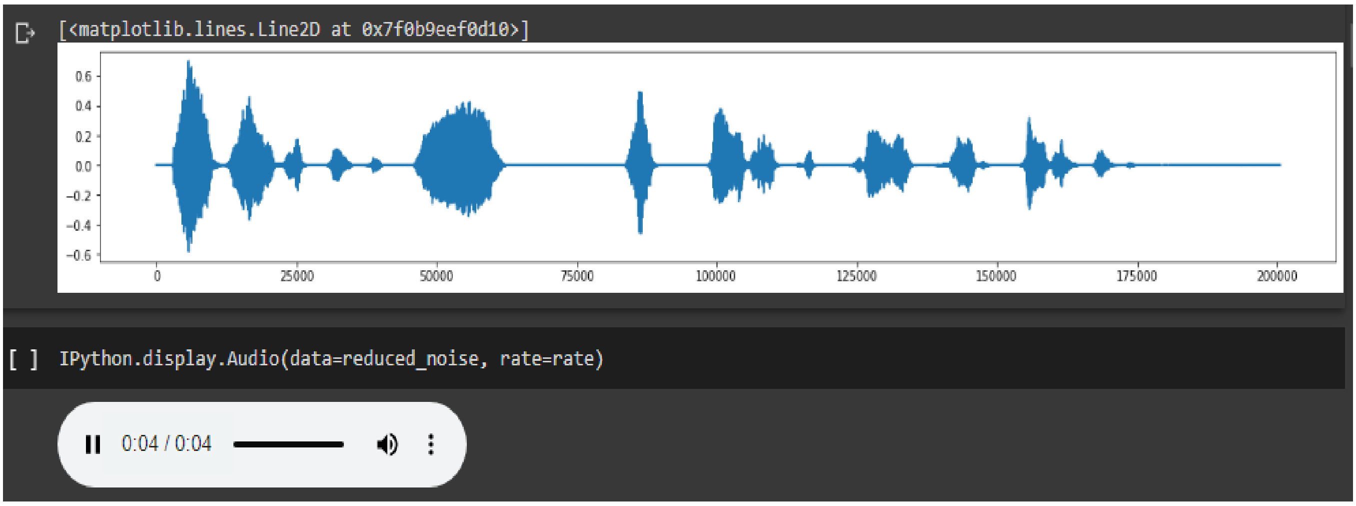
Download the required packages from the python libraries and here I downloaded the required packages Noisereduce, soundfile for inserting the audio clip and reduction using SNR after passing the required Test cases successfully I tested the SNR algorithm using the matplot, noisereduce, soundfile, numpy etc For plotting the audio signals and also for measuring the measurements in the inserted audio file and I inserted one audio sample which is clearly audible and I plotted the frequency graph to it by using the matplot library.



To the audio signal I inserted the noise audio clip and plotted the graph and audio and here we can See the difference before and after adding noise to the audio file



By using SNR reduction of noise we obtained the audio signal without noise



And I explained the project by taking a difficult example by adding some noises into the audio by reducing the noise signal into some batches and displaying the signal before reduction and after reduction of noise.

The software used is snrnoisereduce using Python version in google colab and libraries.

```
%load_ext autoreload
%autoreload 2


#%env CUDA_VISIBLE_DEVICES=3

#......................Download packages if in Google Colab.................

colab_requirements = [
    "pip install librosa",
    "pip install noisereduce",
    "pip install soundfile",]


import sys, subprocess


def run_subprocess_command(cmd):

 # run the command

process = subprocess.Popen(cmd.split(), stdout=subprocess.PIPE)

# print the output

for line  in process.stdout:
        print(line.decode().strip())
IN_COLAB = "google.colab" in sys.modules
if IN_COLAB:
    for i in colab_requirements:
        run_subprocess_command(i)
```

```
#................Test noise reduction algorithm and view steps of algorithm................

import IPython
from scipy.io import wavfile
import noisereduce as nr
import soundfile as sf
from noisereduce.generate_noise import band_limited_noise
import matplotlib.pyplot as plt
import urllib.request
import numpy as np
import io
%matplotlib inline

#.........................Load data............................

url = "https://raw.githubusercontent.com/timsainb/noisereduce/master/assets/fish.wav"
response = urllib.request.urlopen(url)
data, rate = sf.read(io.BytesIO(response.read()))
data = data
IPython.display.Audio(data=data, rate=rate)
fig, ax = plt.subplots(figsize=(20,3))
ax.plot(data)
noise_len = 2 # seconds
noise = band_limited_noise(min_freq=2000, max_freq = 12000, samples=len(data), samplerate=rate)*10
noise_clip = noise[:rate*noise_len]
audio_clip_band_limited = data+noise
fig, ax = plt.subplots(figsize=(20,3))
ax.plot(audio_clip_band_limited)
IPython.display.Audio(data=audio_clip_band_limited, rate=rate)

#.......................Signal to noise ratio noise reduction.............................

reduced_noise = nr.reduce_noise(y = audio_clip_band_limited, sr=rate,
n_std_thresh_stationary=1.5,stationary=True)
fig, ax = plt.subplots(figsize=(20,3))
ax.plot(reduced_noise)
IPython.display.Audio(data=reduced_noise, rate=rate)
reduced_noise = nr.reduce_noise(y = audio_clip_band_limited, sr=rate,
thresh_n_mult_nonstationary=2,stationary=False)
fig, ax = plt.subplots(figsize=(20,3))
ax.plot(reduced_noise)

#....................A more difficult  example............................

url = "https://raw.githubusercontent.com/timsainb/noisereduce/master/assets/cafe_short.wav"
response = urllib.request.urlopen(url)
noise_data, noise_rate = sf.read(io.BytesIO(response.read()))
fig, ax = plt.subplots(figsize=(20,4))
ax.plot(noise_data)
IPython.display.Audio(data=noise_data, rate=noise_rate)
```

```
#.........................Add noise to data...............................

snr = 2 # signal to noise ratio
noise_clip = noise_data/snr
audio_clip_cafe = data + noise_clip
.......................plot noisy data..................................
fig, ax = plt.subplots(figsize=(20,4))
ax.plot(audio_clip_cafe)
IPython.display.Audio(data=audio_clip_cafe, rate=noise_rate)

#.......................Stationary remove noise......................

reduced_noise = nr.reduce_noise(y = audio_clip_cafe, sr=rate, y_noise = noise_clip,
n_std_thresh_stationary=1.5,stationary=True)
fig, ax = plt.subplots(figsize=(20,3))
ax.plot(audio_clip_cafe)
ax.plot(reduced_noise)
IPython.display.Audio(data=reduced_noise, rate=rate)

#......................Non-stationary noise reduction...................

reduced_noise = nr.reduce_noise(y = audio_clip_cafe, sr=rate,
thresh_n_mult_nonstationary=2,stationary=False)
fig, ax = plt.subplots(figsize=(20,3))
ax.plot(audio_clip_cafe)
ax.plot(reduced_noise, alpha = 1)
IPython.display.Audio(data=reduced_noise, rate=rate)
IPython.display.Audio(data=reduced_noise, rate=rate)


#.....................ensure that noise reduction does not cause distortion when prop_decrease == 0...........

noise_reduced = nr.reduce_noise(y=data, sr=rate, prop_decrease=0, stationary=False)
fig, axs = plt.subplots(nrows=2, figsize=(20,6))
axs[0].plot(data[3000:5000])
axs[0].plot(noise_reduced[3000:5000])
axs[1].plot(data)
axs[1].plot(noise_reduced)

#.............................Reduce noise over batches in parallel on long signal...........................

long_data = np.tile(data, 10)
len(long_data)/rate
fig, ax = plt.subplots(figsize=(20,4))
ax.plot(long_data)
noise = band_limited_noise(min_freq=2000, max_freq = 12000, samples=len(long_data), samplerate=rate)*10
audio_clip_band_limited = long_data+noise
fig, ax = plt.subplots(figsize=(20,3))
ax.plot(audio_clip_band_limited)reduced_noise = nr.reduce_noise(
    y=audio_clip_band_limited,
    sr=rate,
    thresh_n_mult_nonstationary=2,
    stationary=False,
    n_jobs=2,
```

```
fig, ax = plt.subplots(figsize=(20,3))
ax.plot(audio_clip_band_limited)
ax.plot(reduced_noise)
reduced_noise = nr.reduce_noise(
    y=audio_clip_band_limited,
    sr=rate,
    thresh_n_mult_nonstationary=2,
    stationary=True,
    n_jobs=2,
fig, ax = plt.subplots(figsize=(20,3))
ax.plot(audio_clip_band_limited)
ax.plot(reduced_noise)
```

#..................Reduce noise on only a subset of a long clip....................

```
from noisereduce.noisereduce import SpectralGateStationary
sg = SpectralGateStationary(
    y = data,
    sr = rate,
    y_noise=None,
    prop_decrease=1.0,
    time_constant_s=2.0,
    freq_mask_smooth_hz=500,
    time_mask_smooth_ms=50,
    n_std_thresh_stationary=1.5,
    tmp_folder=None,
    chunk_size=600000,
    padding=30000,
    n_fft=1024,
    win_length=None,
    hop_length=None,
    clip_noise_stationary=True,
    use_tqdm=False,
    n_jobs=1,
```

#.......................Multichannel noise................

```
audio_clip_cafe_2_channel = np.vstack([audio_clip_cafe, audio_clip_cafe])
audio_clip_cafe_2_channel.shape
reduced_noise = nr.reduce_noise(y = audio_clip_cafe_2_channel, sr=rate,
n_std_thresh_stationary=1.5,stationary=True)
reduced_noise.shape
fig, axs = plt.subplots(nrows= 2, figsize=(20,5))
axs[0].plot(audio_clip_cafe_2_channel[0])
axs[1].plot(audio_clip_cafe_2_channel[1])
```

```
axs[0].plot(reduced_noise[0])
axs[1].plot(reduced_noise[1])
IPython.display.Audio(data=reduced_noise, rate=rate)
reduced_noise = nr.reduce_noise(y = audio_clip_cafe, sr=rate, thresh_n_mult_nonstationary=2,stationary=False)
reduced_noise.shape=fig, ax = plt.subplots(figsize=(20,3))
ax.plot(audio_clip_cafe)
ax.plot(reduced_noise, alpha = 1)
IPython.display.Audio(data=reduced_noise, rate=rate)
```

6. Final version of prototype / product (only images):

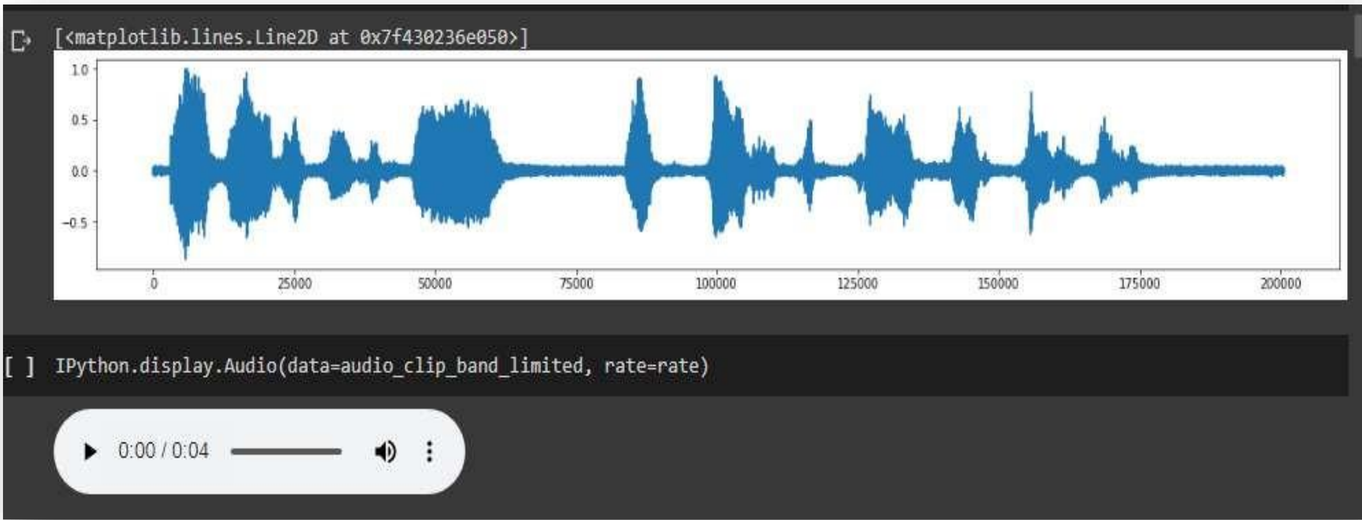


Figure 1audio signal without any noise

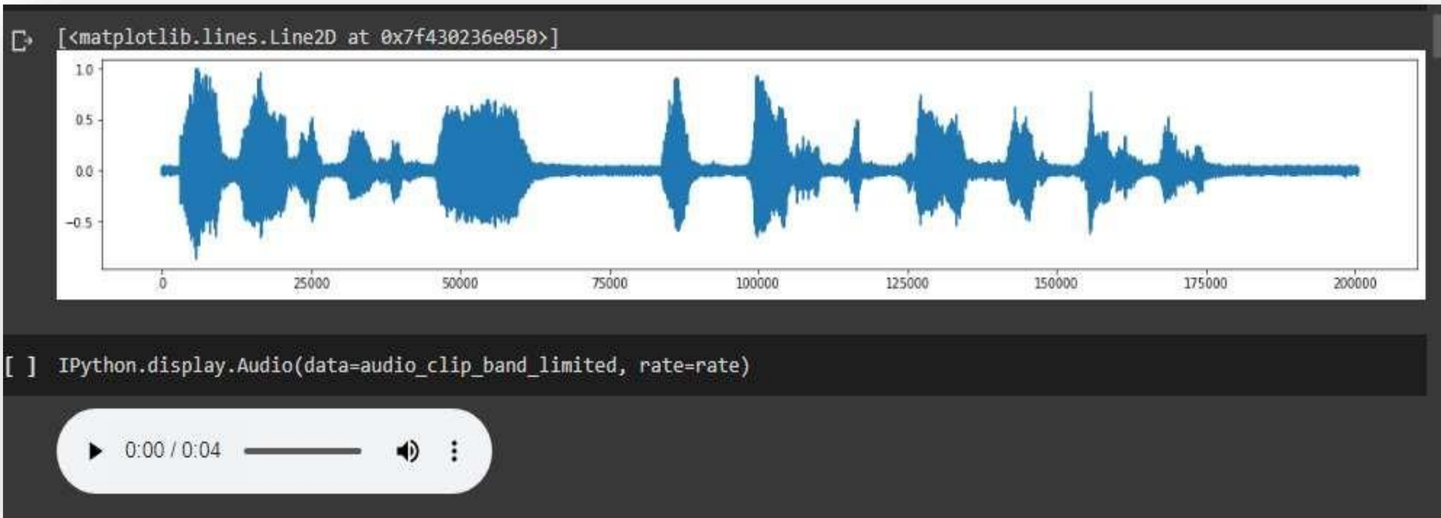


Figure 2 Audio signal after inserting noise

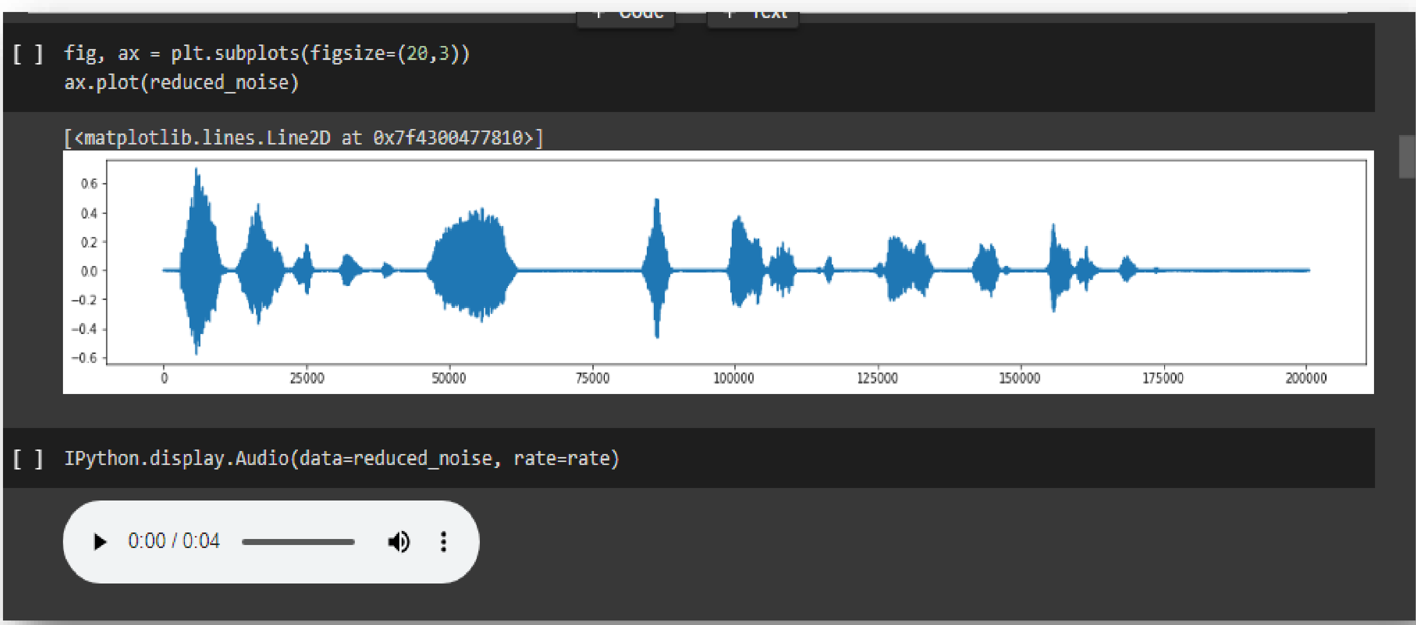


Figure 3 audio signal after reducing the noise

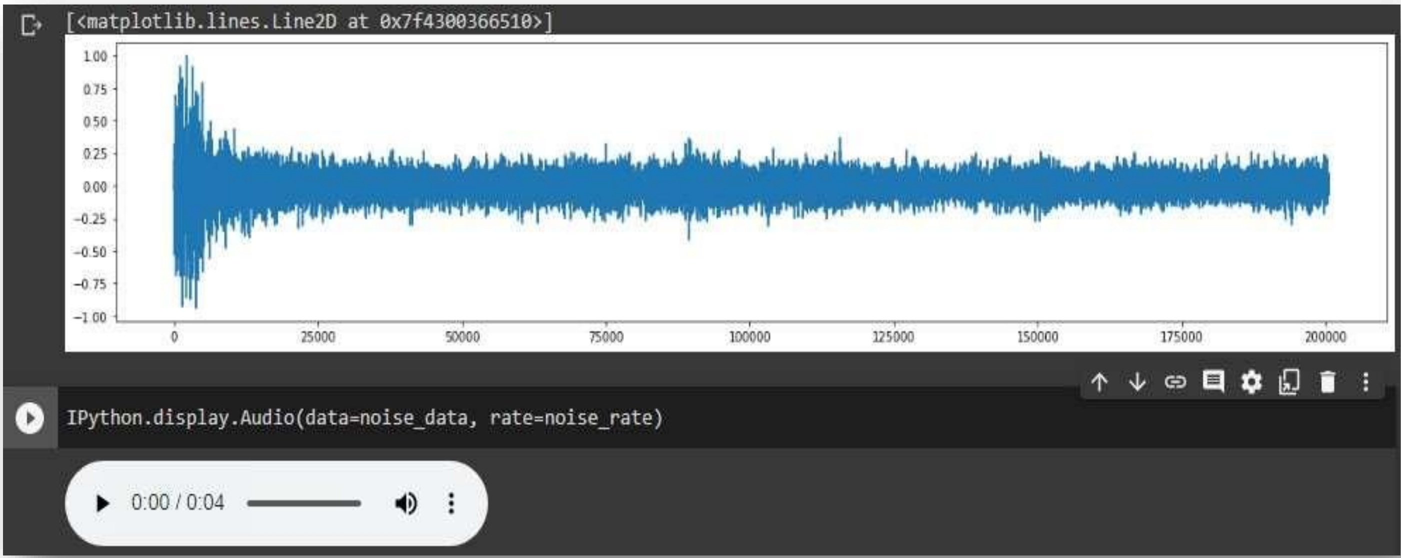


Figure 4 little complicated example

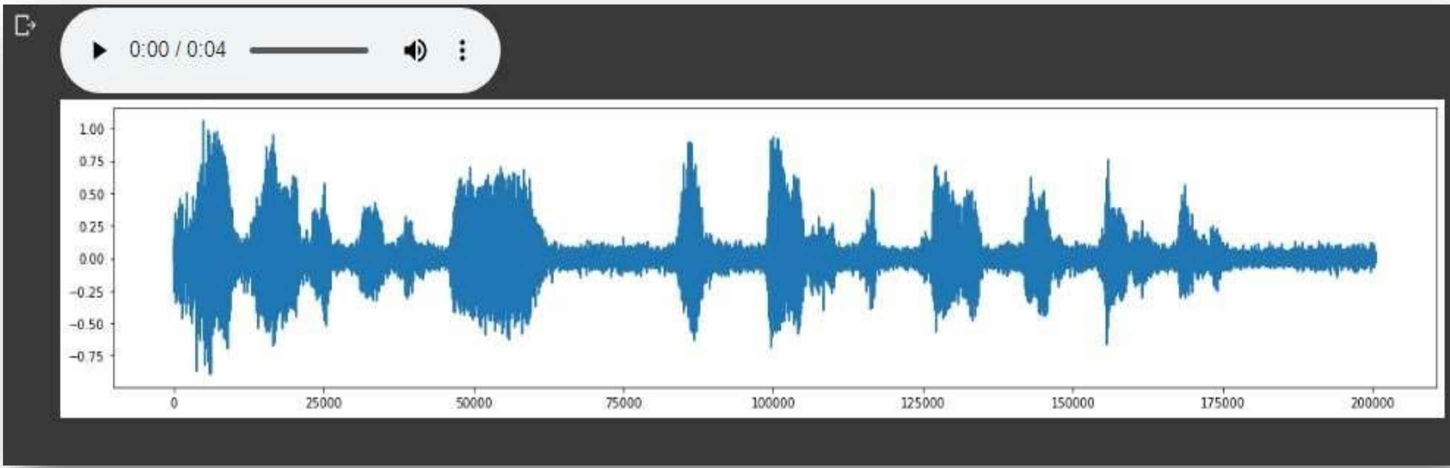


Figure 5 non stationary noise reduction

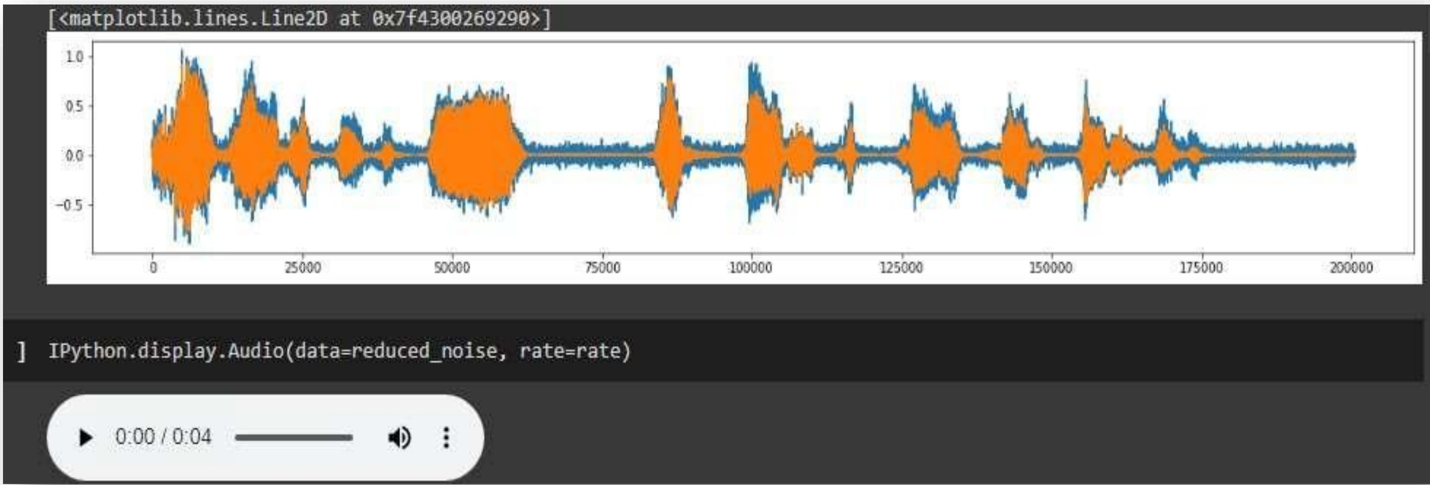


Figure 6 stationary noise removing

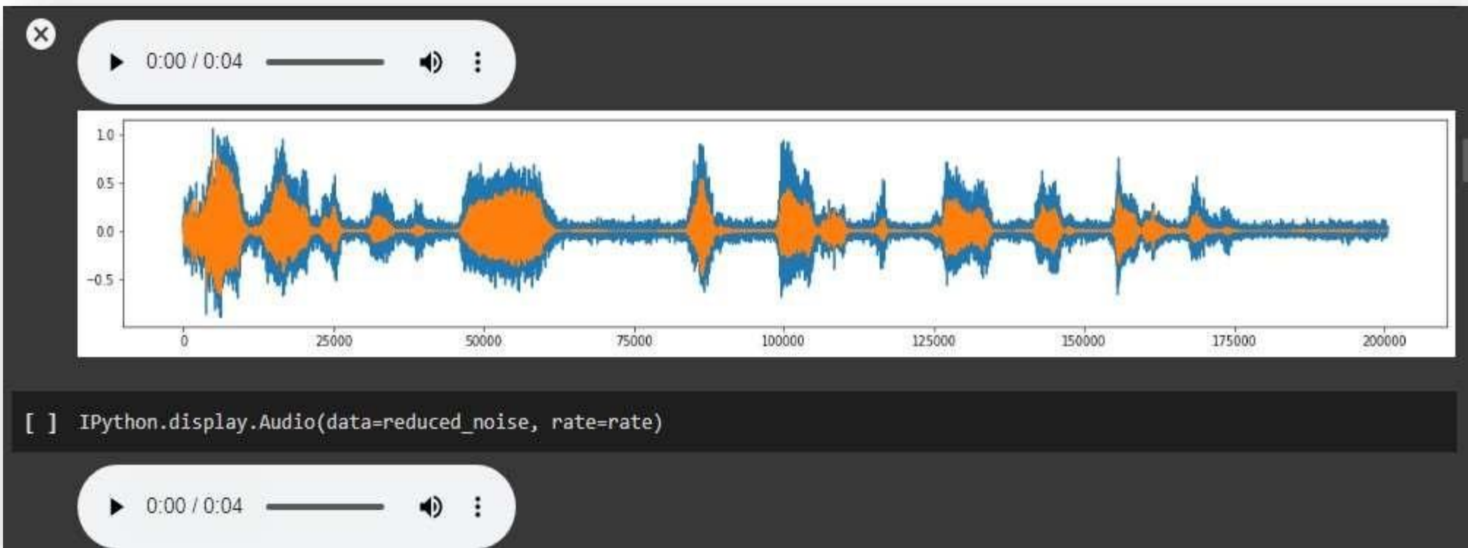


Figure 7 Non-stationary noise reduction

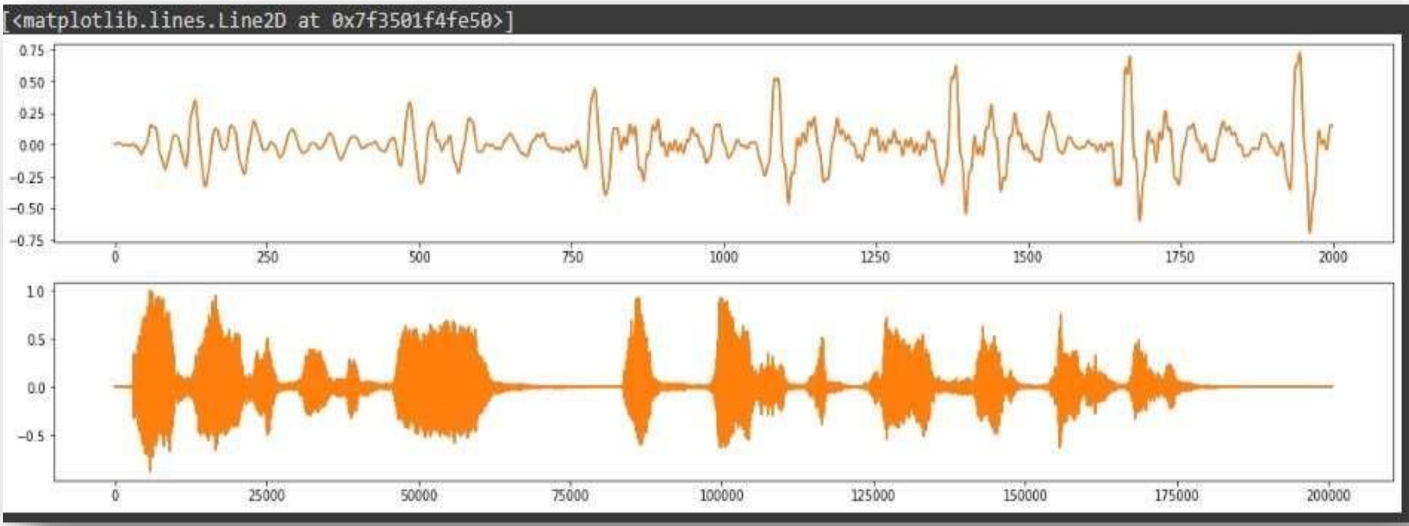


Figure 8 ensuring that noise reduction does not cause distortion

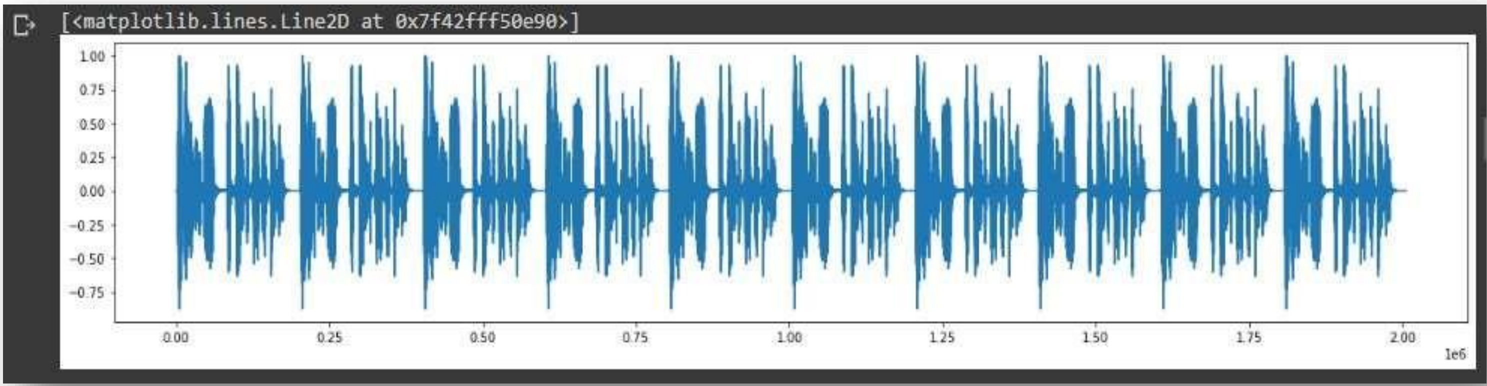


Figure 9 reduce noise over batches in parallel on long signal

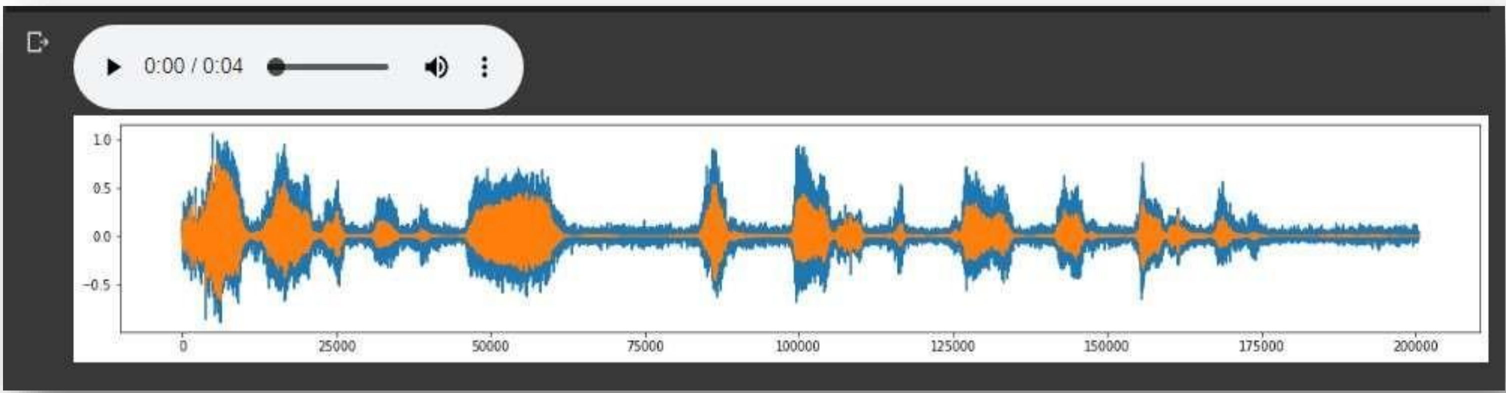


Figure 10 Signal after removing multi channel noise

1. **Any other information**:

The method which is used to reduce noise can be implemented in telecommunications for better Enhancement of noise cancellation using SNR filter signal-to-noise ratio compares a level of signal power to a level of noise power. It's most often expressed as a measurement of decibels (dB). Higher numbers generally mean a better specification since there's more useful information (the signal) than unwanted data (the noise).

Elevated noise levels can lead to hearing loss and other ear-related issues as well as cause emotional stress that results in diminished worker productivity.

Dean-CLET **Dean -TIIC**