Signal Processing Project

Team Name -: Mixed Signals

Team Members:

Abhishek Sharma (2022102004)

Abhinav S (2022102037)

Himanshu Yadav (2022102010)

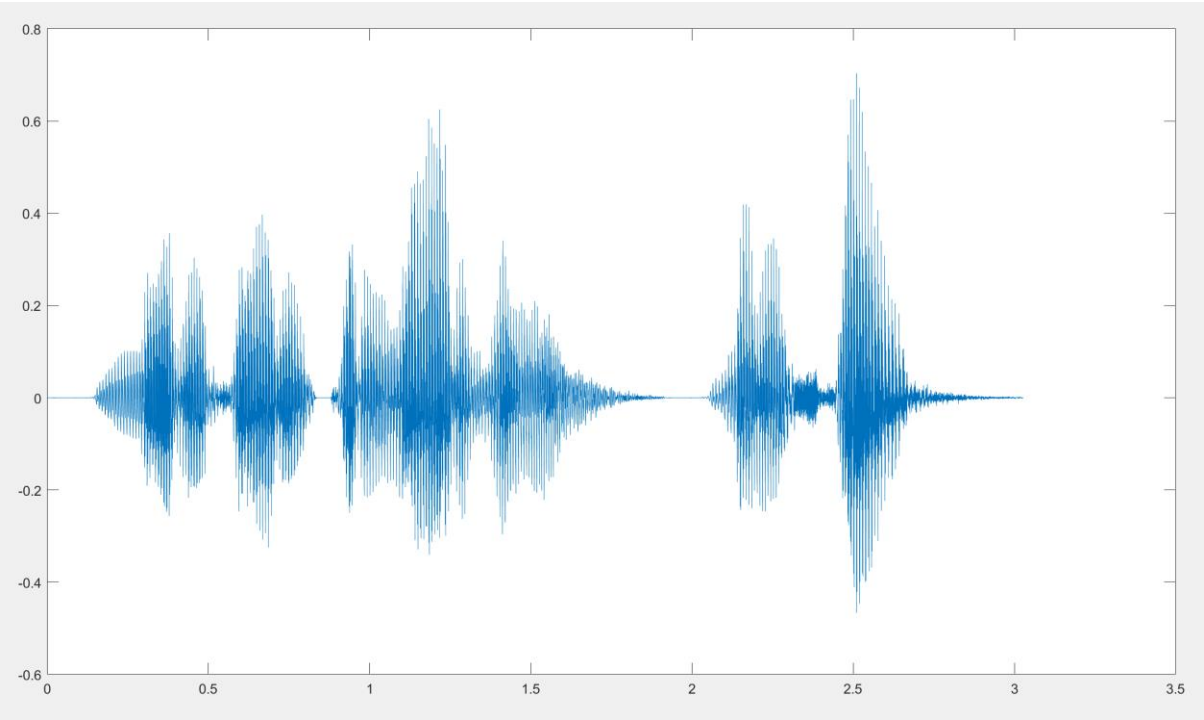
PART – 1: ECHO CREATION

AIM:

Given an input audio signal, create an echo for that sound using signal processing techniques.

INPUT:

Input is taken to be a speech signal, so that echo perception is easier for human ears and thus verifying the output becomes straightforward.



PROBLEM SOLVING APPROACH:

For creating an echo, the original sound is added with a delayed and attenuated version of itself. Theoretically, reflection of sound is considered to be echo only if there are no multiple reflections. This echo creation can be implemented using an FIR filter. However, practically the reflections of sound are continuous, and they die out when the attenuation reaches a certain level. The phenomenon of multiple sound reflections is called reverberation. Which can be implemented using an IIR filter.

FIR FILTER:

For a single attenuated reflection signal we can use the following difference equation to create echo:

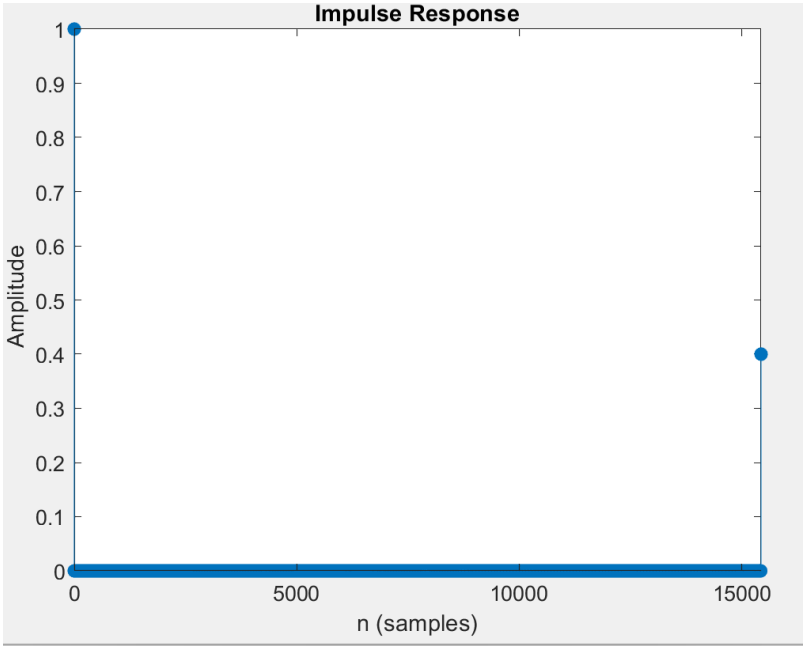
$$y[n] = x[n] + a.x[n - n_0]$$

Here a is the attenuation of sound on each reflection and n_0 is the delay after which echoed signal starts playing. n_0 can be calculated based on the delay in seconds given by the equation,

$$n_0 = Delay(in\ s) * F_s$$

The delay for echo can be calculated using the distance between reflecting surface and speed of sound easily. However, the distance for echo to occur is usually small since air obstruction and other obstacles cause echo to attenuate.

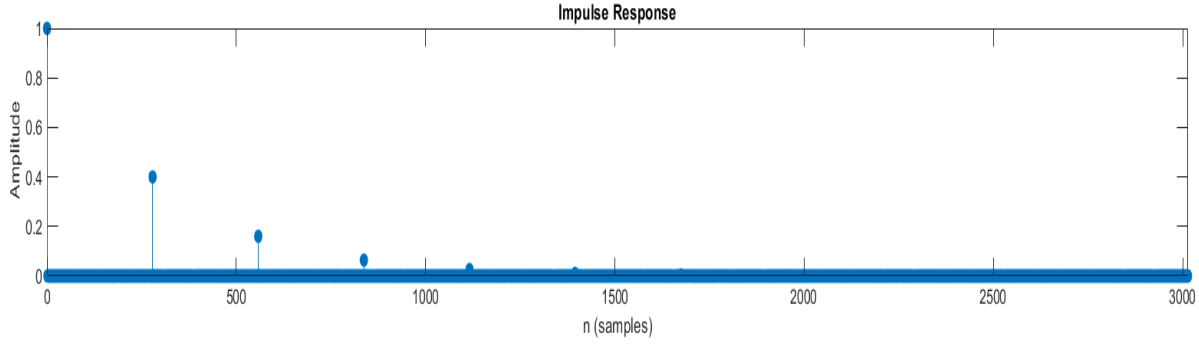
The impulse response for the filter above is,



IIR Filter: For creating multiple reflections which is naturally occurring too, the difference equation we can use is,

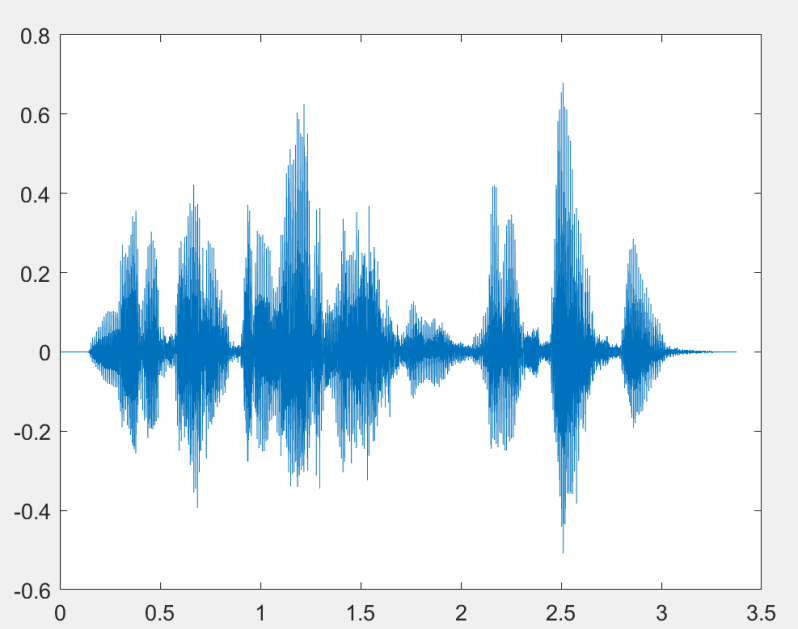
$$y[n] = x[n] + a.y[n - n0]$$

The Impulse response for above filter is as below,

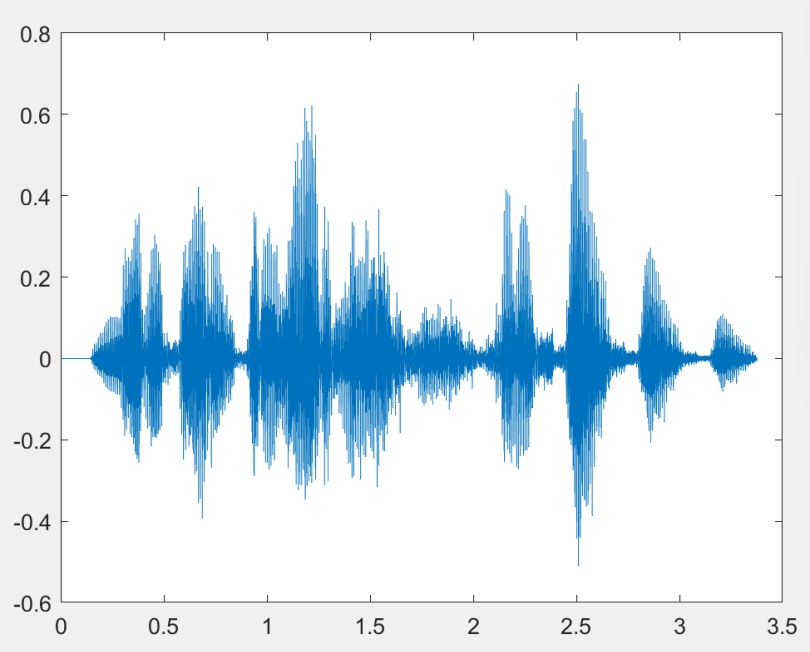


RESULTS:

Echo created using single reflection gives the following output:



Echo created using multiple reflection which is the echo in natural setting(reverberation)



CONCLUSION:

Thus, we can conclude that we can create an echo effect in any given audio signal using a simple filter corresponding to difference equations mentioned above.

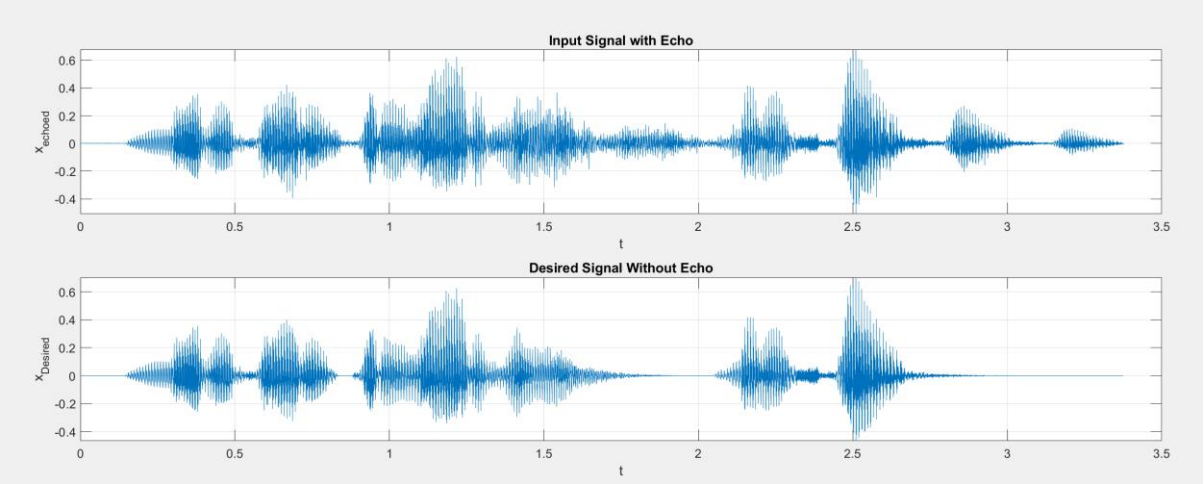
PART – 2: CANCEL THE ECHO

AIM:

Given the desired audio signal as well as an audio signal which contains echo effect, cancel the echo effects from signal and produce an output with no echo.

INPUT:

The desired audio signal and another with the original audio and echo with non-uniform delays.



PROBLEM SOLVING APPROACH:

Adaptive Filter has been used to remove the echo from the signal and produce the desired output.

For implementing the adaptive filter, LMS (Least Mean Square) algorithm has been used. LMS algorithm adapts the filter by changing the filter coefficients, to minimize least mean square error between the output signal of the filter and the provided desired signal.

Let x[n] be the input signal which has echo, d[n] be the desired signal, w[n] be the filter coefficients of the adaptive filter. We want to adapt the filter such that the output y[n] of the adaptive filter resembles the desired signal d[n] as much as possible.

In order to do this, we move through the input signal x[n] by taking windows of length equal to the order of the adaptive filter, suppose N. At a time-instant n, let the samples considered under the window, be represented by signal z[n]. So, z[n] passes through the filter and produced output y[n]. Now we calculated the error in the output by subtracting it from the desired output, let's say that

error is e. Then the filter coefficients for the next iteration will be updated by the following formula.

$$w[n+1] = w[n] + 2\mu.e.z[n]$$

Where w[n+1] is the filter tap weights for the next iteration, w[n] is equal to the filter tap weights for the current iteration, μ is a constant, z[n] is a part of the original signal in the current iteration having length equal to filter tap weight vector.

In order to do this, we move through the input signal x[n] by taking windows of length equal to the order of the adaptive filter, suppose N. At a time instant n, let the samples considered under the window, be represented by signal z[n]. So, z[n] passes through the filter producing output y[n] at time instant n.

$$y[n] = \sum_{i=0}^{N-1} (w(i)x(n-i))$$

The error signal of the output with respect to the desired signal at time instant n is:

$$e[n]=d[n]-y[n]$$

We have to adapt the filter coefficients such that second moment of e[n] is getting minimized.

For this, at each instant, the filter coefficients are adapted.

$$w(n+1)=w(n)+2\mu e(n)z(n)$$

where,

w[n] is the filter coefficents at time instant n

w[n+1] is the filter coefficients at time instant n+1

e[n] is the error at time instant n

z[n] is the the signal under consideration at time instant n.

μ is a constant.

The various parameters to be selected include:

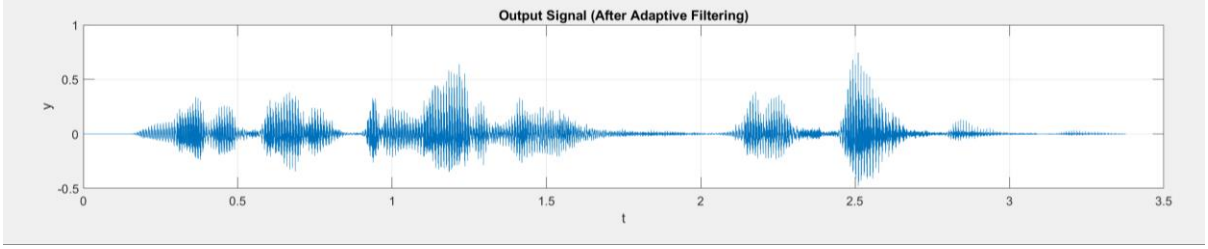
1. 1.μ(step size):The step size decides the rate of convergence and stability of the adaptation of the filter.Too small a value makes the adaptation stable and divergence will not occur, but adaptation will be slow and might require multiple iterations to get the desired output.Too large a value, will ensure a faster adaptation, but might lead to

divergence and unstability of the system especially, when the audio environment is unstable.

2. 2.Filter order:The filter order depends on the input as well as the computational complexity our system can handle. If the input signal is predictable(less fluctuations), then, a lower filter order might be sufficient to record it.But if the input signal is random and has too many fluctuations, then a larger filter order will produce the desired output, but this will put a strain on the system, demanding too many computations.

In the code, μ has been taken to be 0.014, small enough to ensure stability and filter order to be 300. The filter coefficients have been initialized to be 0.

RESULTS:



CONCLUSION:

Upon listening, we observe that, in the output signal, the echo has been removed, almost to the point that it can't be perceived by our human ear. Though upon careful inspection, we note that the error is more initially, before adaptation starts as compared to later. Thus, given an audio with echo and desired signal, the echo can be removed successfully using adaptive filters.

PART – 3: WHAT IS THIS NOISE?

AIM:

To accurately identify and categorise the type of noise present in the recording, distinguishing source of origin among Fan, Pressure Cooker, Water Pump and City Traffic.

INPUT:

A music recording which contains a noisy audio signal where noise is one of the above-mentioned types.

PROBLEM SOLVING APPROACH:

Use of MFCC (Mel Frequency Cepstral Coefficients) has been approached to solve the problem. For each of the noises, MFCC has been found and analysed. Based on the observations, simple statistical values like Mean, Root Mean Square Error have been found and used as features for classification.

Short Time Fourier Transform

When we compute DFT of a discrete time signal x[n],we can find the frequency components in the signal. We can find what frequency components are dominant but not when .So, we don't have information about the temporal characteristics of the signal.

In STFT, we divide the signal into small segments of some specific window length for which the audio remains stationary and compute its DFT locally. For speech, this is taken around 20-30ms which is the minimum time between two glottal closure. The DFT can be computed efficiently using algorithms like FFT.

This is done by taking frames throughout the signal considering some jump size typically half of the window size. The overlap between frames allows us to keep track of the characteristics of the signal.

Before computing DFT, windowing is done in order to make the signal narrow towards the ends of the frame.

Considering all these steps, the STFT can be represented as:

$$S(m,k) = \sum_{n=0}^{N-1} x(n+mH).w(n).e^{-\frac{i2\pi kn}{N}}$$

where x[n] is the input signal, m is the frame number under consideration, H is the jump size, N is the window size and w(n) is the windowing function.

Typically, windows used are Hanning or Hamming window.

While taking DFT we have to consider N such that N is greater than the window size, so as to make sure that no information is lost in this step.

The number of frames in total is given by:

$$frames = \frac{samples - framesize}{hopsize} + 1$$

Where samples refers to total number of samples in x[n].

Mel Filter Banks

Variation in frequency of audio is not perceptible to human ear in Hz scale. We can differentiate between 300Hz and 310Hz but we can't differentiate between 2000 and 2010Hz both of which have the same absolute difference between them. For this, we can use a different scale where our human ear can perceive the absolute difference between them. This is called Mel scale.

Formula for interconversion between Hz and Mel scale is as follows:

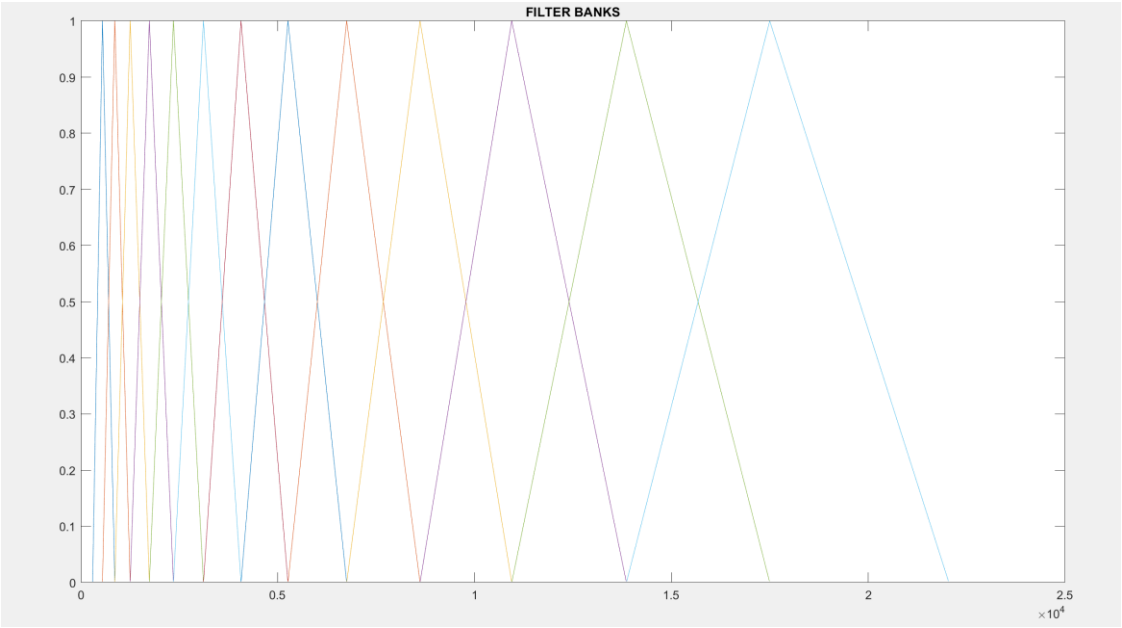
$$f = 100.\,(10^{\frac{m}{2595}} - 1)$$

$$m = 2595.\log\,(1 + \frac{f}{500})$$

where m is frequency in Mel, f frequency is Hz.
Mel filter banks consists of triangular filters. The range of frequencies present in the filter banks are non-linear in Hz, whereas linear in Mel.

Steps in designing Mel Filter Banks include:

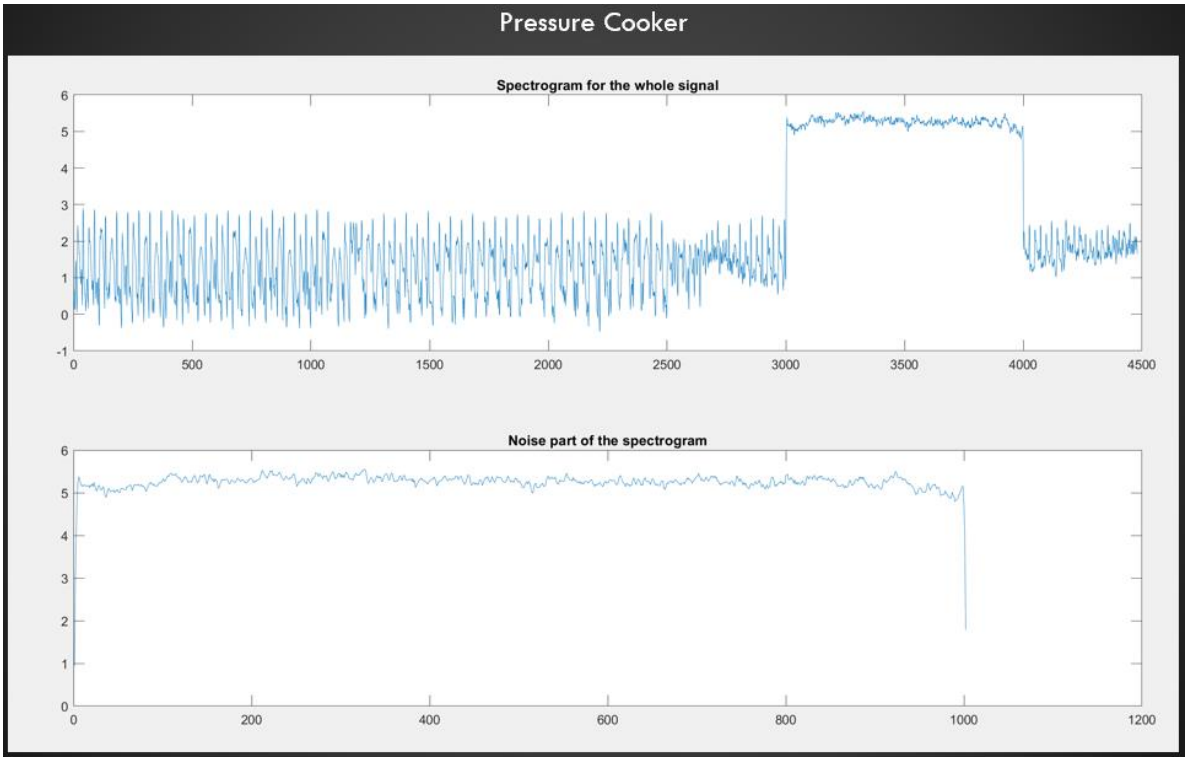
1. While Designing Mel Filter Banks, we have to decide the number of triangular filters we will be considering. Typically K=13 triangular filters are considered. Then, K equally spaced frequencies are taken in Mel scale between the lowest and highest frequencies(half of sampling frequency).The highest is considered as such as we assume signal is real. Therefore, power spectrum is symmetric. These linear spaced frequencies in Mel scale is converted to Hz.
2. Filter banks are constructed between them as shown in the diagram.



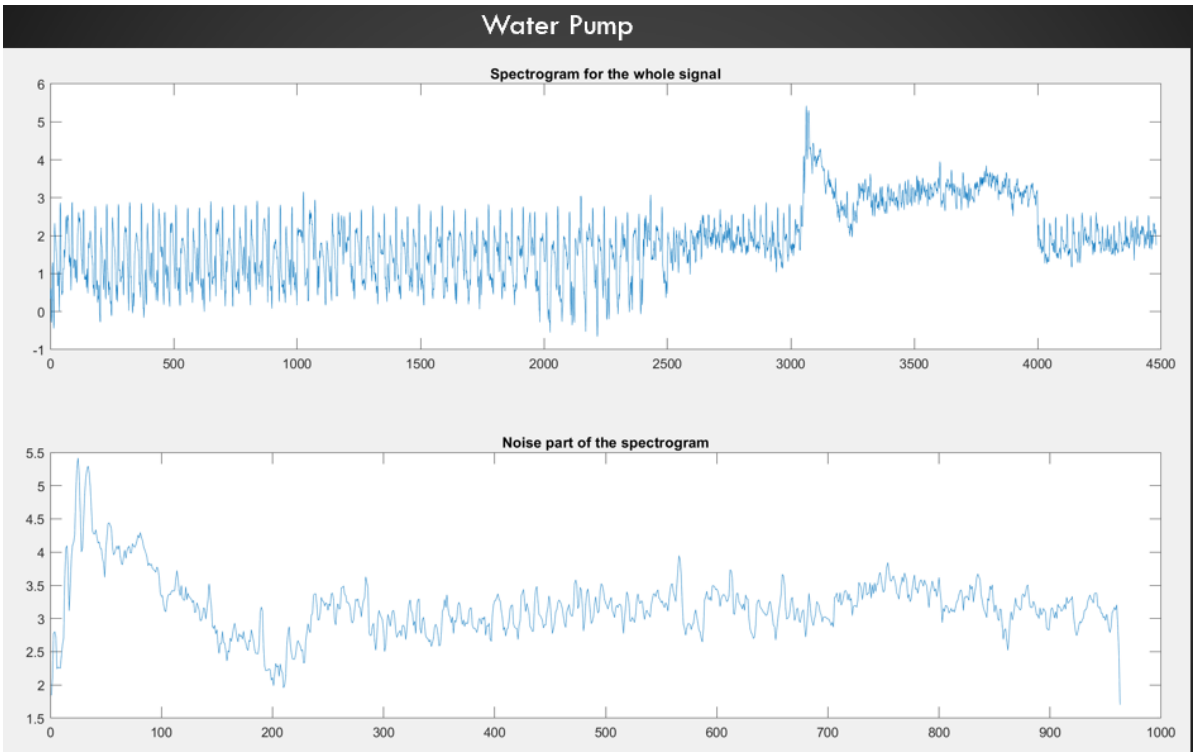
3. The Power spectrum of the signal is passed through the designed Mel filter bank to get the K MEL bands. Each of the Mel bands can be considered to be the strength of the range of frequencies in it in the input signal across time.

Upon analyzing the bands at the time where noise is heard in the audio, observations about the noise are made. We observed that all the noise is recorded in the same band and the different characteristics of it used for their classification are discussed in the next section.

NOISE CHARACTERISTICS:

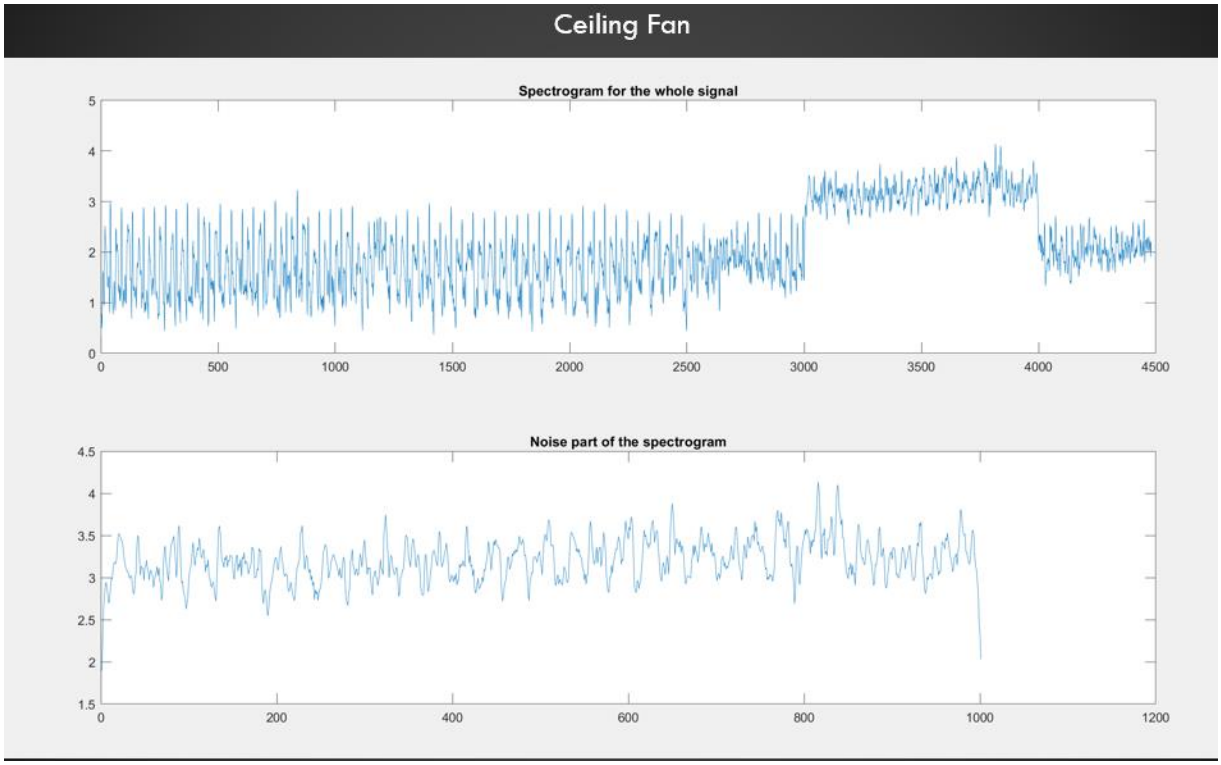


Since the noise in case of pressure cooker lies in a very small frequency range the rms error from maxima and Variance of the noise will have very small values compared to the rest.



Since the maxima is at a very high frequency level from the general noise range in case of water pump, the rms error from maxima is very high.

For the traffic noise, there are horn noises after some delays and thus the noise level varies a lot hence, the variance will be higher in comparison to ceiling fan.



Since, ceiling fan noise is again restricted to a smaller frequency range than in case of city traffic noise, thus, the variance will be lower.

CONCLUSION:

This approach to noise classification seemed to work as we can find in which Mel band the noise belongs to and analyse it across time to classify it. The speech or music will not interfere with the result as we can define the bands such that they belong to another band which makes classifying the noise easier.