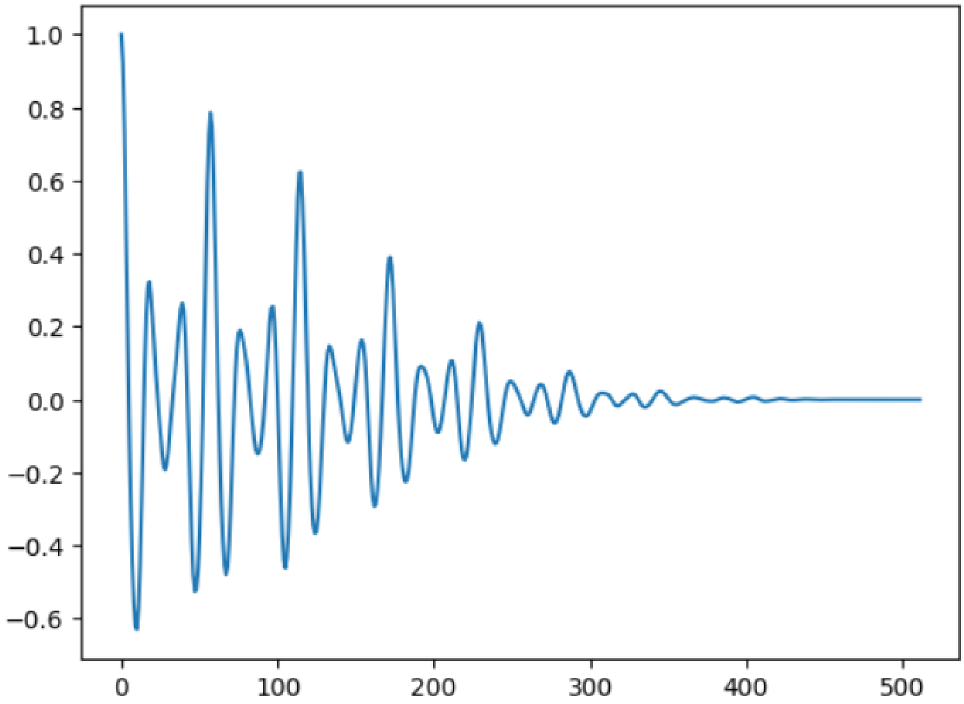
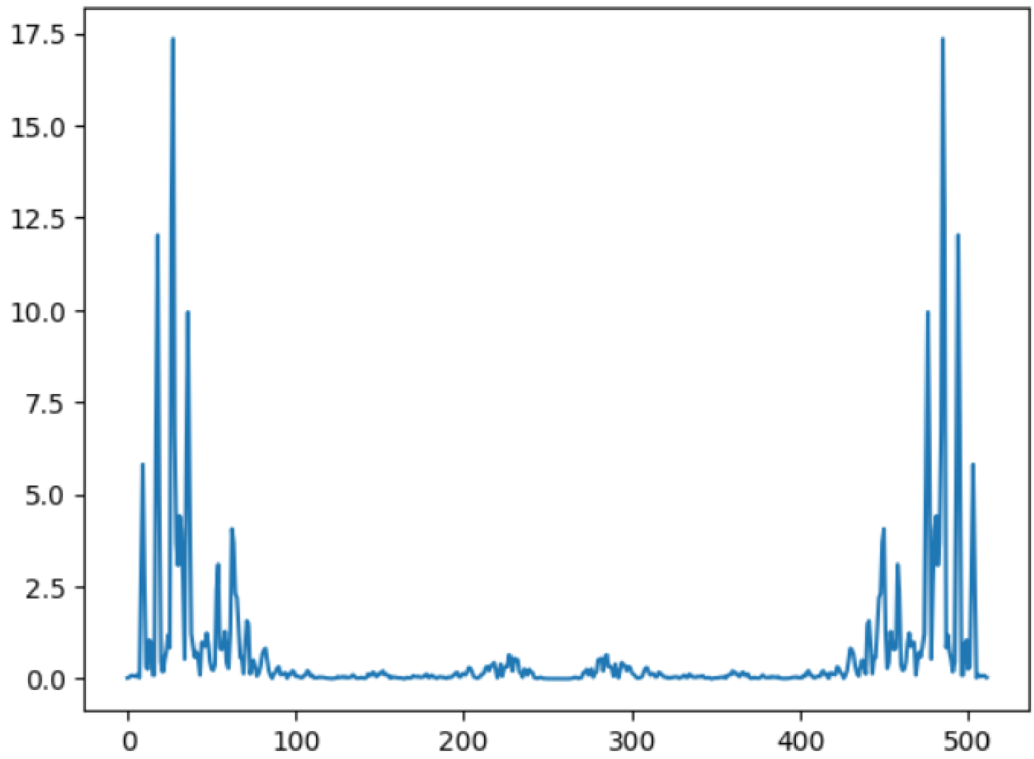
1.a) short term auto correlation

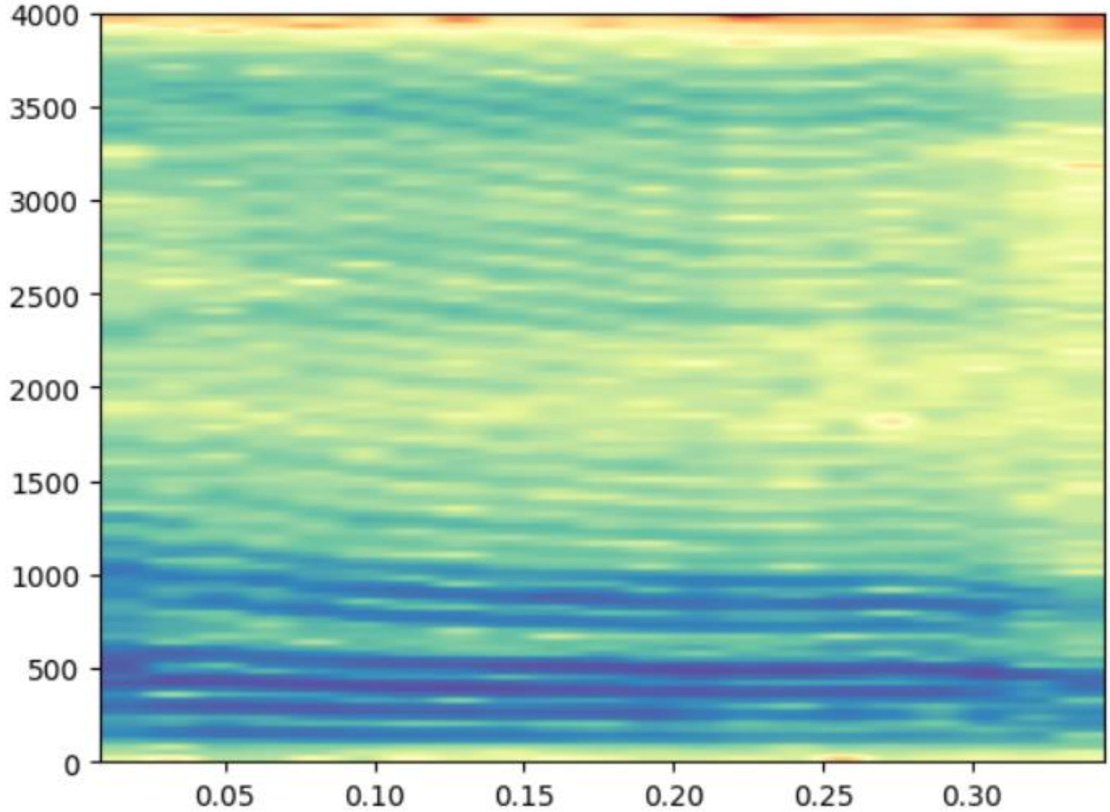
[<matplotlib.lines.Line2D at 0x1ffce6d1060>]




1.b) Magnitude spectrum



3.V**oiced:**



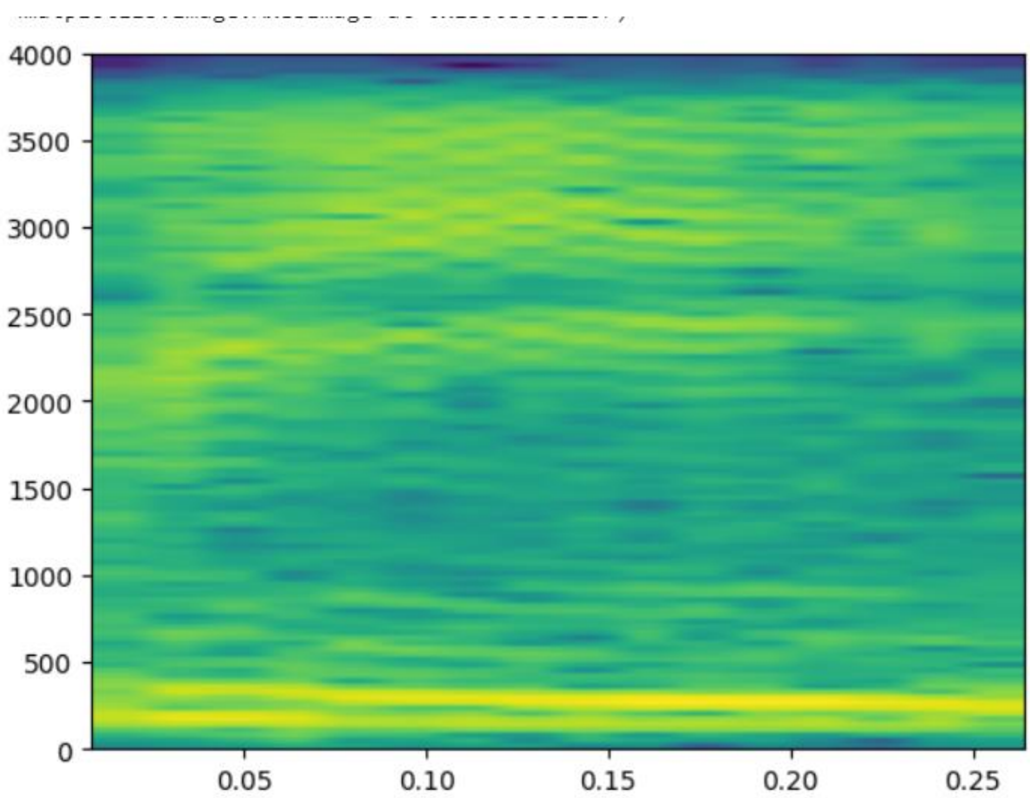

A formant is dark band on spectrogram which tells information about the vocal tract shape & resonance. They are frequency which have high degree of energy.

Here vowel o is taken for plotting spectrogram for voiced. Formant 1 and formant 2 are very close and appear as single formant band. That is there no much difference of gap between two formants. Formant2 is at 1000 hz and formant is at 500 hz.

The first formant is inversely related to vowel height. That is higher the formant frequency and lower the tongue height. Here formant is pretty much high, so it is mid vowel (o)

The second formant is related to degree of backness. That is more front the vowel, the higher the second formant. Here formant is very low, so it is a back vowel.

## Unvoiced:

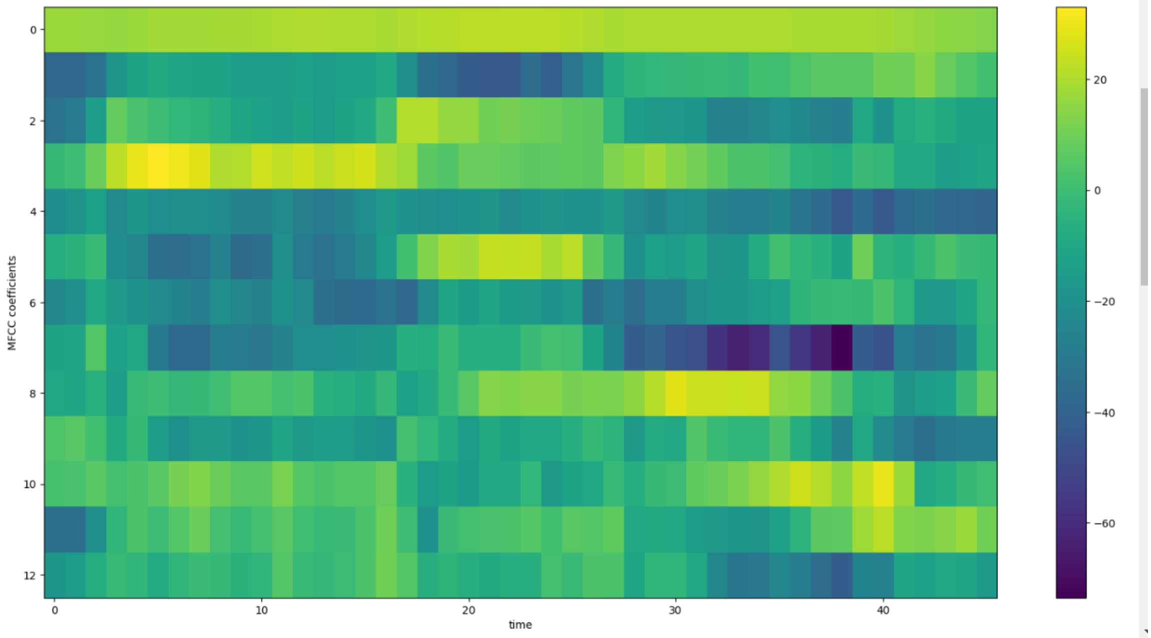



Plosives (/p/), for example, are defined by a period of complete silence, and by how they "bend" the vowels around them. It does not give any information on formants

Unvoiced sounds does not have consonants since vocal cords will not vibrate during the production of consonants. So formants does not exist for unvoiced

2) MFCC

```
[9]: <matplotlib.colorbar.Colorbar at 0x1ebb87fd510>
```




MFCCs are the Mel Frequency Cepstral Coefficients. MFCC takes into account human perception for sensitivity at appropriate frequencies by converting the conventional frequency to **Mel Scale.**

Wherever the cepstral coefficient has a positive value, that indicates the spectral energy is concentrated in the low-frequency regions. That is voiced sounds which has lower frequency.

1$^{st}$ coefficient of all frames seems to be high

On the other hand, if a cepstral coefficient has a negative value, it represents that most of the spectral energy is concentrated at high frequencies.

For the frame ranging from 30 to 40 has some negative values which indicates high frequency components. In that range we can say that fricatives(/sh/) is present.

**Estimate the pitch using the two autocorrelation results. Which result would provide better performance in an autocorrelation procedure?**

Pitch using autocorrelation will give better performance since it gives the peaks in periodic manner which gives repeating pattern. So it is easy to calculate the pitch using autocorrelation. This method is efficient for both periodic and quasi-stationary. But using magnitude spectrum is good only for periodic signals.

**Comment the changes in both the autocorrelation and the spectrum. What do these changes indicate about the effects of the clipping operations on the waveform?**

If we take autocorrelation without windowing we will get the below waveform. But when we apply the windowing and taking autocorrelation will give the waveform asymptotically decreasing. Windowing will reduce the sidelobes and eliminate the spurious data.

In magnitude spectrum 2nd half of the frequencies is conjugate symmetric to the first half.

Here first N/2 output bins are typically the only bins that you we are interested in. The first bin is DC (0 Hz), and bin N/2 corresponds to Nyquist (Fs/2). Bins above N/2 are just complex conjugate "mirror images" of the bottom N/2-1 bins.

4)spectral subtraction

```
>> q4
Signal to noise ratio before enhancement:
   -7.3174

Signal to noise ratio after enhancement:
   -5.8287

fx >>
```

SNR ratio is reduced after spectral subtraction

**References:**


1. (867) Python Tutorial: Learn Scipy - Fast Fourier Transform (scipy.fftpack) in 17 Minutes - YouTube
2. Python Examples of python_speech_features.mfcc (programcreek.com)
3. numpy.hamming — NumPy v1.23 Manual
4. Find Peaks in Python | Delft Stack
5. MFCC's Made Easy. An easy explanation of an important… | by Tanveer Singh | Medium