Real-time Pitch estimation

Shrey Mehta

***Abstract*—The task is to take as input a real-time audio signal and to output the pitch of the signal in real-time as the audio signal is fed.**

There are two good ways in which the task of pitch estimation can be carried out:

- Auto-correlation
- Fast Fourier Transform (FFT)

We could have used the auto-correlation method to estimate the pitch but if the input audio signal provided is $\{x[n]\}$, then to know the value of auto-correlation at that time, we would also need the values of the signal from $x[n+1]$ to $x[n+l]$, where l is the window size. But, we want to estimate the pitch in real-time alongside the feeding of the signal, so we use the FFT method to estimate the pitch of the audio provided.

We will follow the following steps to estimate the pitch of the audio signal:

1) Take the input signal $\{x[n]\}$ as input (use pyaudio)
2) **Calculate the FFT**: Apply the FFT on $\{x[n]\}$ to get X[k] (use scipy.fft())

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi}{N}kn}$$

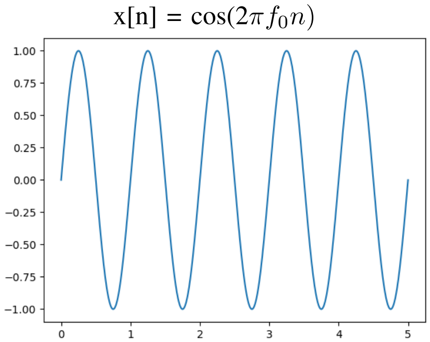
3) **Magnitude Calculation**: At this instant, calculate the magnitude of the FFT result by taking the absolute value of each complex element. So, an array of magnitudes is obtained. (use numpy.abs())
4) **Peak Identification**: To find spectral peaks, we locate local maxima in the magnitude spectrum and get the index of it from the array of magnitudes, say $peak\_index$ (use numpy.argmax())
5) **Convert Indices to Frequencies**: Once you've identified the indices of the spectral peaks, you can convert these indices to corresponding frequencies (use scipy.fftfreq())

This will give us the desired frequency (**f**) of the signal at the given instant of time.

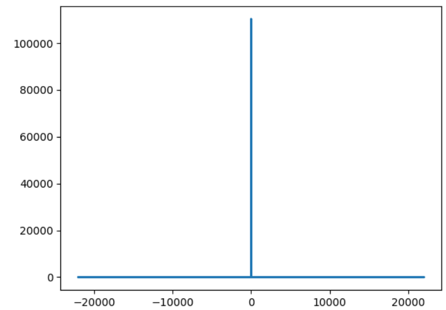
Example:

Consider the input signal to be a sinusoidal wave of the form

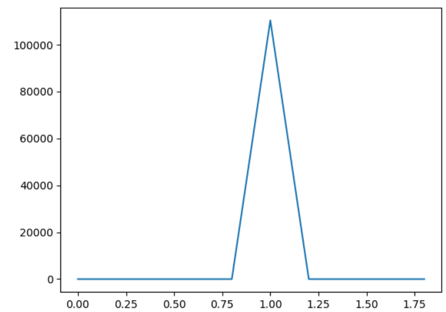
$$x[n] = \cos(2\pi f_0 n)$$



We apply the FFT on this input signal and we obtain the result the following plot of FFT magnitudes to the spectrum frequencies:



On zooming in around 0, we get the following plot:



So, we get the maximum value of the FFT magnitude (N/2) when the frequency is 1 Hz, so we can conclude that the value of f_0 is 1.

Image processing: Since the input signal may be of any kind (i.e. may contain random noise or may not be expressible in terms of sinusoids), we can multiply the input signal $x[n]$ with a smoothening window function, like the Hann window or the Gaussian window. It has been seen Hann window is used commonly for pitch detection, so we also use this for our pitch estimation purpose.

$$w[n] = 0.5\left(1-\cos\left(2\pi n/(N-1)\right)\right)$$

So, before finding out the FFT in step 2, we can smoothen the audio signal by using

$$xs[n] = x[n]w[n]$$

Then, we can calculate the FFT as

$$X[k] = \sum_{n=0}^{N-1} xs[n]e^{-j\frac{2\pi}{N}kn}$$

So, this method of using the peaks of the FFT is useful in estimating the pitch of the signal in real time as

$$\text{Pitch} \propto \mathbf{f}$$

Hence, we have estimated the pitch of an audio signal in real time.