Individual Project

Abdul Fatah Jamro

December 22, 2022

Contents

1 Introduction

Noise in the data has remained a major concern for scientists and engineers. Various techniques are used like shields to protect or avoid noise, as well as denoising techniques. Till now, Fast Fourier Transform (FFT) has remained on of the best approach to denoise, compress or analyse the data. In the field of signal processing, noise removal is a basic issue which is tackled FFT. It gets a lot easier when FFT is done with the help of computing tools. In this study Python is used as tool to apply Fast Fourier transform in denoising a noisy audio signal. Following sections briefly explain Fourier series, Fourier transform, and Fast Fourier transform, followed by example coding in Python.

1.1 Fourier Series

A wave form or any periodic function is represented as a Fourier series, which is the sum of sines and cosines. It bears the name Jean-Baptiste Joseph Fourier after the French mathematician and scientist (1768–1830). Fourier transformations are frequently utilized in signal processing because sine waves are the building blocks of sound waves [1]. Complex waves, like sound, can be said as a combination of sine waves using a Fourier series. The series adds together the sines and cosines of a wave.

This implies that a wave's constituent parts can be separated from one another. The study of various Fourier series falls within the category of Fourier analysis. A Fourier transform is a technique for separating a signal into its various frequencies [1]. Although a Fourier series can be calculated manually, computers are better able to decipher the intricate harmonics of commonplace sounds. In audio processing, such as when isolating specific sounds from a recording, Fourier analysis is frequently utilized [1].

1.2 Fourier Transform

The Fourier transform (FT) uses pattern based on time as its input to calculate the strength, rotation speed, and total cycle offset for each potential cycle. Waveforms, having a function of time, space, or another variable, are subjected to the Fourier transform. The Fourier transform represents any signal into sinusoid form.

A time function waveform is broken down into its constituent frequency function using the Fourier transform mathematical function. The Fourier transform generates a complex valued function of frequency as its output. While the Fourier transform's complex argument indicates the phase offset of the fundamental sinusoidal in that frequency, its absolute value represents the frequency value present in the original function.

Fourier transform is referred as a generalization of the Fourier series. FT can be used to describe both the mathematical operation used and the representation of the frequency domain. With the help of FT, the Fourier series can also be applied to non-periodic signals

The Fourier Transform (FT) of a function $f(x)$ is:

$$F(k) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i k x} \, dx \tag{1}$$

and the inverse can be obtained using inverse Fourier transform:

$$f(x) = \int_{-\infty}^{\infty} F(k) e^{2\pi i k x} \, dk \tag{2}$$

Some of the properties of Fourier transform are as follows.

Linear transform: If $g(t)$ and $h(t)$ are two Fourier transforms given by $G(f)$ and $H(f)$ respectively, then the Fourier transform of the linear combination of g and t can be easily calculated.

Time shift property: The Fourier transform of $g(t-a)$ where a is a real number that shfts the original function has the same amount of shift in the magnitude of the spectrum.

Modulation property: A function is modulated by another function when it is multiplied in time.

Parseval's theorem: Fourier transform is unitary, i.e., the sum of square of a function $g(t)$ equals the sum of the square of its Fourier transform, $G(f)$.

Duality: The Fourier transform of $G(t)$ is g if $g(t)$ possesses the Fourier transform $G(f)(-f)$.

2 Fast Fourier Transform (FFT)

The discrete Fourier transform (DFT) is a tool to do some sorts of functions of sequences into some other kind of representations. DFT is long computing method whereas FFT is fast and short, relatively. A fast Fourier transform (FFT) algorithm calculates the discrete Fourier transform (DFT) of some sequence.

The discrete Fourier transform also converts a waveform's cycle structure into sine components, which is another way to explain it. A Fast Fourier transform can be applied to many different signal processing techniques. It might be helpful for image-processing technology or for reading things like sound waves. A Fast Fourier transform can be used to quickly solve different kinds of equations or display different kinds of frequency activity.

Fast Fourier transform and the DFT, which are exceedingly technical aspects of both computing and electrical engineering, are mostly the domain of engineers and mathematicians attempting to alter or build components of various technologies. For example, fast Fourier transform might be helpful in sound engineering, seismology or in voltage measurements. The FFT is a crucial measurement technique in the study of measuring audio and acoustics. It breaks down a signal into its distinct spectral components, giving frequency information about the signal in the process.

FFTs are used for machine or system condition monitoring, quality control, and fault analysis. This page covers the operation of an FFT, the pertinent parameters, and how they affect the measurement outcome. The FFT is efficient implementation of the DFT.

A signal is separated into its frequency components after being sampled over time. Each of these elements is a discrete sinusoidal oscillation with a unique frequency, amplitude, and phase. The figure that below shows the transformation.

The Fourier Transform is a potent tool for looking at data from the frequency domain, as unlike to the time domain. With its mathematical formulas, this powerful process, however, appears frightening. Transform time-domain wave to frequency-domain, where N is the number of samples:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi ikn/N} \tag{3}$$

Data can be easily manipulated in frequency domain, like removing noise or applying data compression. Then, we can reverse the frequency domain to time domain by reversing equation

$$x_n = \sum_{k=0}^{N-1} X_k e^{2\pi ikn/N} \tag{4}$$

Figure 1 illustrates the Fourier Transform (FT): decomposition of a sophisticated wave into different sinusoidal waves.

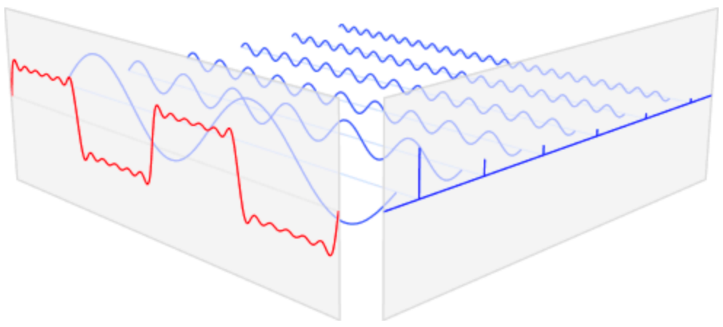

Figure 1: Frequency and time domains of a signal.

Let us put aside for the time being the difficulty of FT equations. Let us pretend that we fully get the meaning of the mathematical equations and apply the Fourier Transform to carry out some useful work in Python [2]. We create two audio signals of different frequencies with the help of python coding as shown in Figure 2 and Figure 3 and convolute them into single resultant signal named clean signal as shown in the Figure 4.

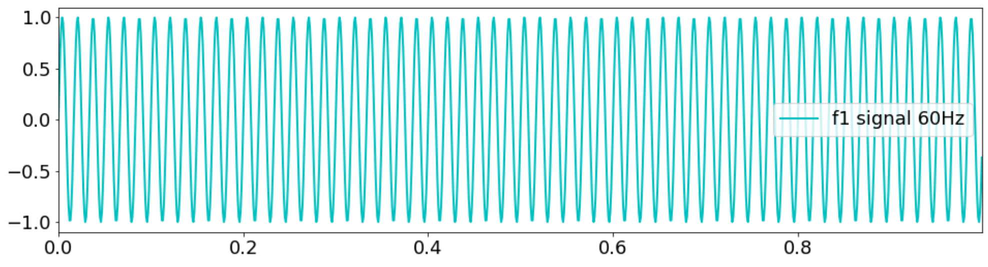

Figure 2: A 60Hz signal.

Figure 2 was generated using the Python code in Listing 1.

```python
import numpy as np
dt = 0.001
t = np.arange(0.0, 1.0, dt)
f1 = np.sin(2.0 * np.pi * 60.0 * t)
plt.plot(t, f1, label='60Hz Signal')
plt.legend()
plt.show()
```

Listing 1: Generating a 60Hz signal.

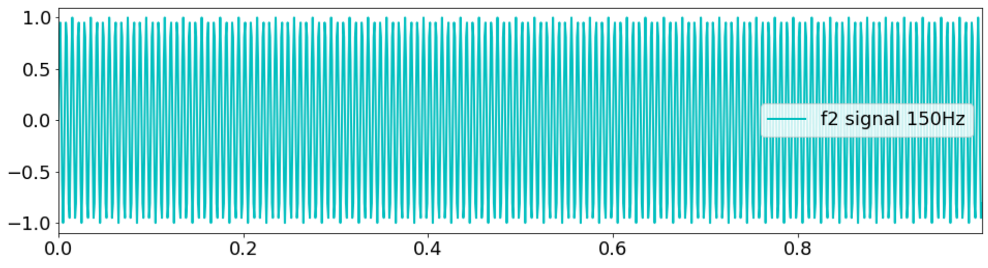

Figure 3: A 150Hz signal.

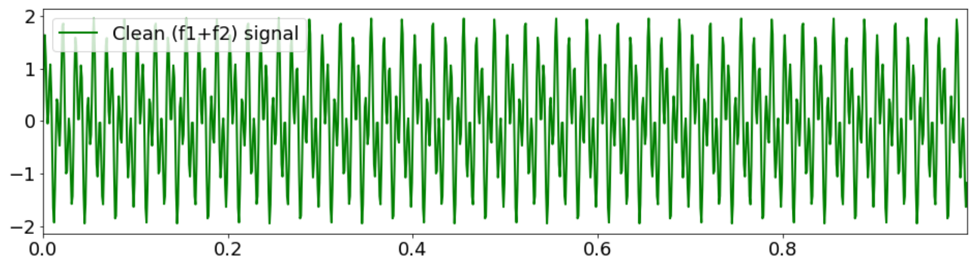

Figure 4: Sum of 60Hz and 150Hz signals.

Then, we deliberately created a random noise signal with random frequency and imposed on our clean audio signal as in Figure 5 and Figure 6, respectively.

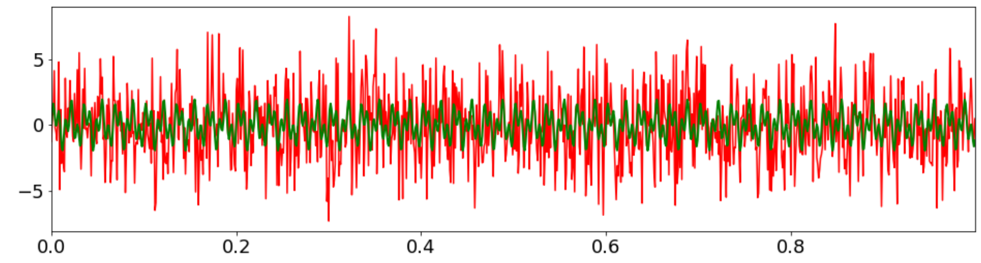

Figure 5: The noisy signal with original superimposed.

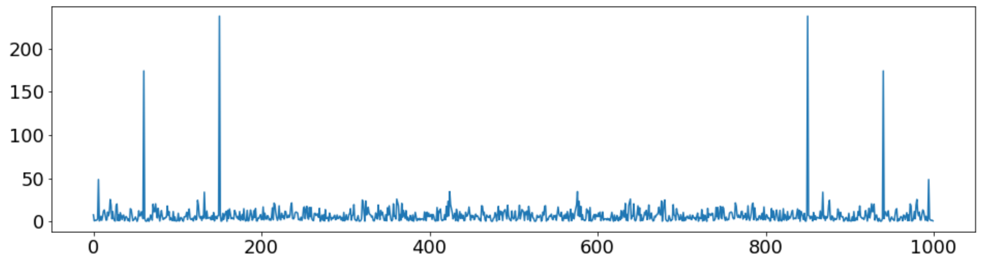

Figure 6: The Frequency domain of the noisy signal.

Above, in Figure 6, is the noisy audio signal that needs to be filtered-out of noise. Here is the use of Fast Fourier Transform (FFT) to filter out the noise and get original signal back.

Initially, the time domain signal is changed into frequency domain using algorithm we already discussed of; FFT algorithm. With the help of Fourier Transform it becomes easier to track frequencies and break signal into different frequencies. FFT algorithm is the efficient way to do that job.

References

[1] G. Tolstov and R. Silverman, *Fourier Series*, ser. Dover Books on Mathematics. Dover Publications, 1976. [Online]. Available: https://books.google.ie/books?id=XqqNDQeLfAkC

[2] "Clean Up Data Noise with Fourier Transform in Python," https://towardsdatascience.com/clean-up-data-noise-with-fourier-transform-in-python-7480252fd9c9, accessed: 2022-12-22.