Lab Report 3 : PHY407

Fadi Farook

October 25, 2024

Fourier Transform

The Fourier transform is the one of the most fundamental mathematical tools used in physics, engineering and signal processing. For any signal in time $f(t)$, we can extract its frequency components using the formula:

$$\hat{f}(k) = \int_{-\infty}^{\infty} f(t) e^{-i2\pi kt} dt$$

When working with experiments or simulations, we find that data is often discrete. Thus, with a finite sequence of values y_n totaling N values, we can retrieve the discrete frequency components:

$$c_k = \sum_{n=1}^{N-1} y_n e^{-i\frac{2\pi kn}{N}} \tag{1}$$

In most practical applications, the Fast Fourier Transform (FFT) [1] is used, which is a more efficient algorithm that provides the same result. Finally, to retrieve the original signal (usually after modification), we perform the inverse:

$$y_n = \frac{1}{N} \sum_{n=1}^{N-1} c_k e^{i\frac{2\pi kn}{N}} \tag{2}$$

In this convention, the inverse Fourier transform divides by N to preserve the amplitude after both operations are performed.

1 Audio Filtering

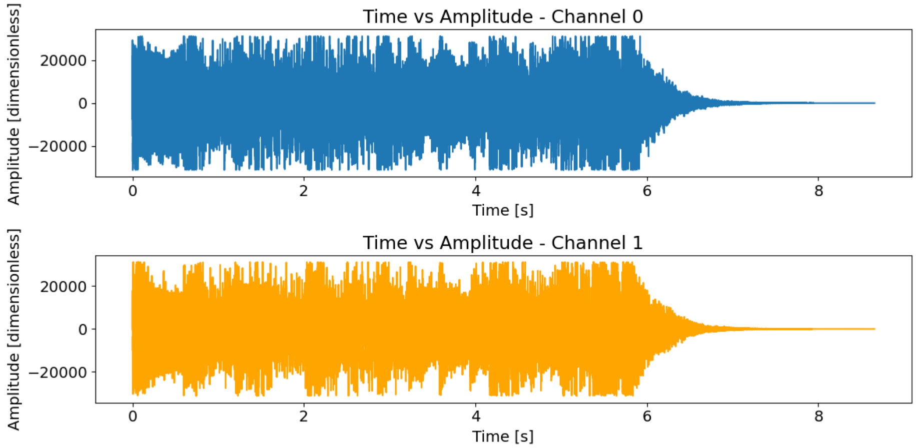

Figure 1: Time vs Amplitude plot of an audio file containing two channels.

A common usage of Fourier transforms is in signal processing, particularly with audio files. An audio signal is usually heard and hence represented as amplitude versus time. Given a sampling rate of f_s, we can relate the nth data point to the time by $t_n = \frac{n}{f_s}$. For example, Fig. 1 shows two channels of an audio file converted to a time versus amplitude plot with a sampling rate of 44100 Hz and 381,700 data points.

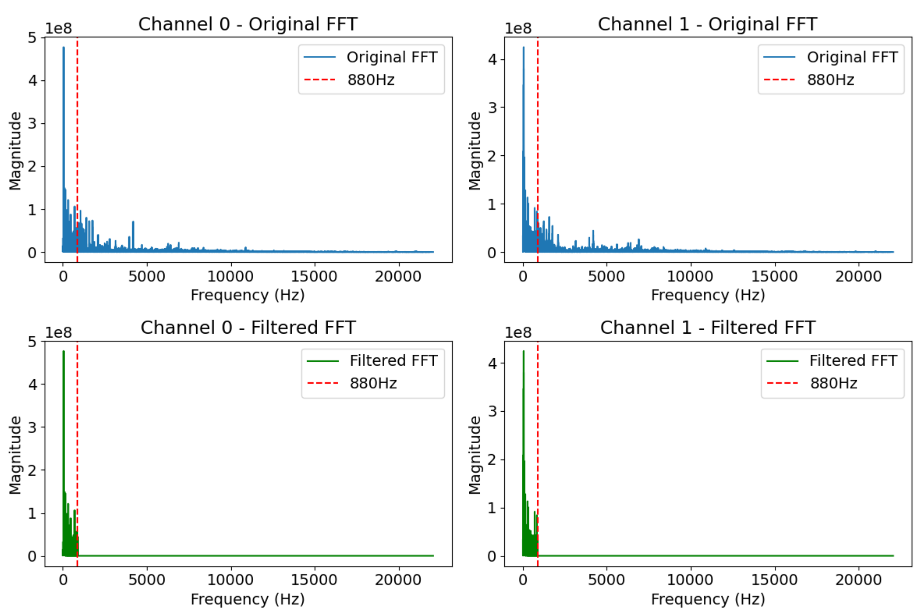

Figure 2: Plots of the frequencies that composed each channel signal, and their magnitudes. Filtered plots also shown after filtering out frequencies above 880 Hz.

Using Equation 1, we can identify the frequencies that make up this signal. A significant advantage of this transformation is the ability to reduce noise within the signal. In this case, we set all frequencies above 880 Hz to zero, as shown in Fig. 2.

The result of this noise reduction, along with a zoomed-in view of a 30 ms segment, is displayed in Fig. 3 and 4. We clearly observe that the fluctuations within millisecond scales have disappeared.

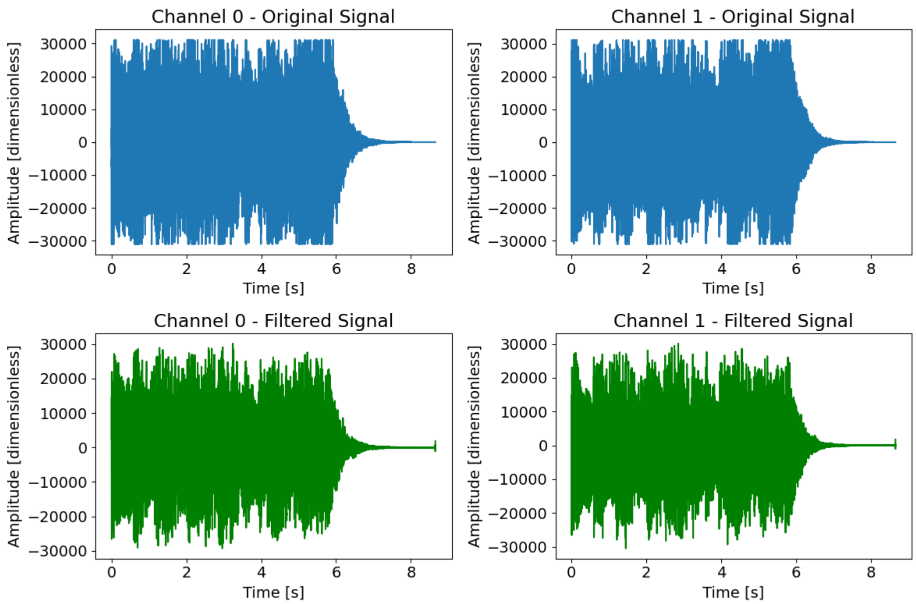

Figure 3: Comparison of signal amplitudes before and after removing 880+ Hz frequency components.

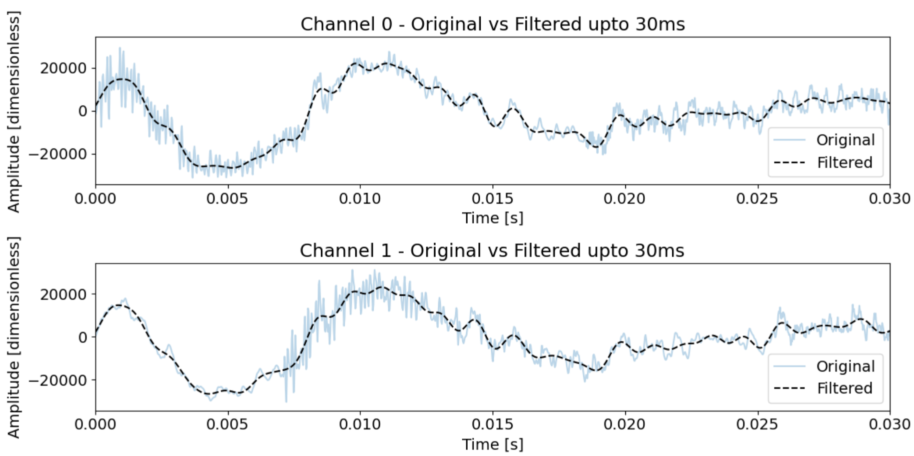

Figure 4: Comparison of signal amplitudes before and after removing 880+ Hz frequency components. Same as Fig. 3, zoomed in from 0 to 30 ms. Reduction of noise is clearly observed.

2 Getting Rich on the Stock Market

This method of signal filtering and noise reduction is not exclusive to sound. Another convenient application is the analysis of stock performance without the influence of short-term volatility. Fig. 5 shows the opening value of the S&P 500 Index on every business day between late 2014 and 2019. Here, we see that it fluctuates significantly in the short term.

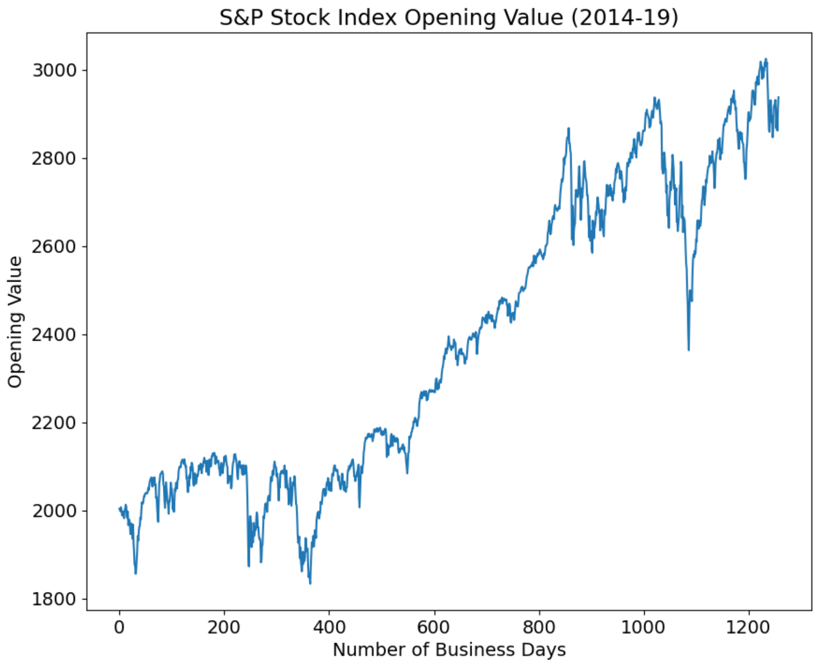

Figure 5: Opening Value of S$P 500 stock index, over 4-5 years. Overall growth is seen with significant short-term volatility

To analyze the long term trends, we can filter out all the frequency components that corresponds to variations less than 6 months. If we assume 4-week months, this corresponds to $t_{cutoff} = 120$ days. In frequency space, this corresponds to 0.008333333 days^{-1}, when using the calculation

$$f_{cutoff} = \frac{1}{t_{cutoff}}$$

The magnitudes at frequency above f_{cutoff} were set to 0. Fig. 6 shows the opening values after applying this filter. We see that it is comparatively much smoother and reflects the long term behaviors of the S&P 500. Due to its close ties to the overall economy, we can garner that these reflect the overall growth in the economy overtime. It also shows periods of recessions (periods of low economic activity over months), bear and bull markets.

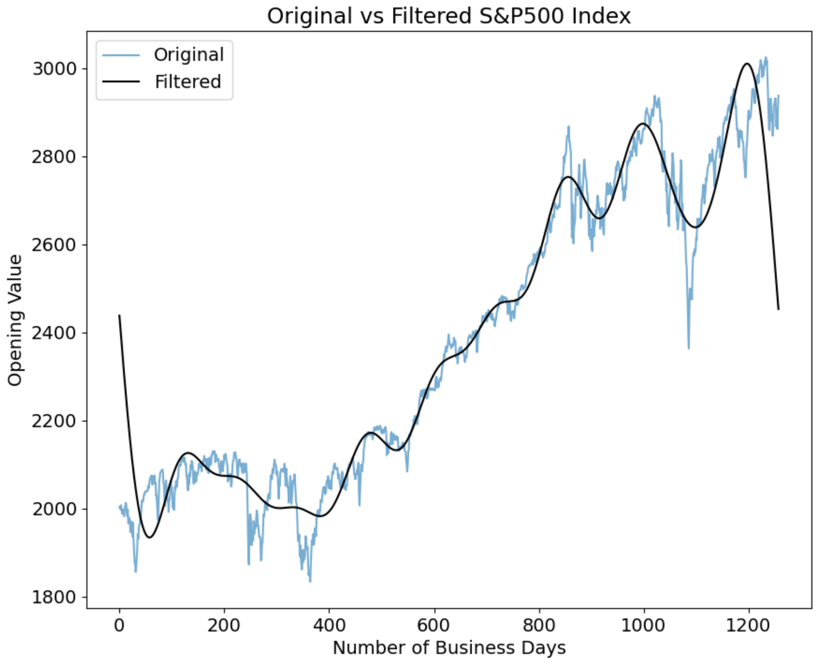

Figure 6: S&P 500 index values as in Fig. 5, overlaid with the same without variations less than 6 months. Clear overall growth, with periods of occasional decline, is seen.

2.1 A Note on Accuracy of FFT

When we Fourier transform this data using NumPy's rfft function [2], we must note that FFT prefers an even number of values, ideally 2^n. This is because the FFT recursively divides the data into two, and combines them back [1].

When returning to the original values by applying rfft and irfft, this impacts our data containing 1259 points. The transformation returns an array of length 1258. However, if the first value is ignored and 1258 values are used, no discrepancies are observed.

In the case of an even array length, the amplitudes were compared. The amplitude is preserved when both transforms are applied, as shown in Fig. 7, which indicates that any differences are due to round-off errors for opening values of order 10^3. The Fourier transform algorithm is remarkably accurate, and does not show approximation error.

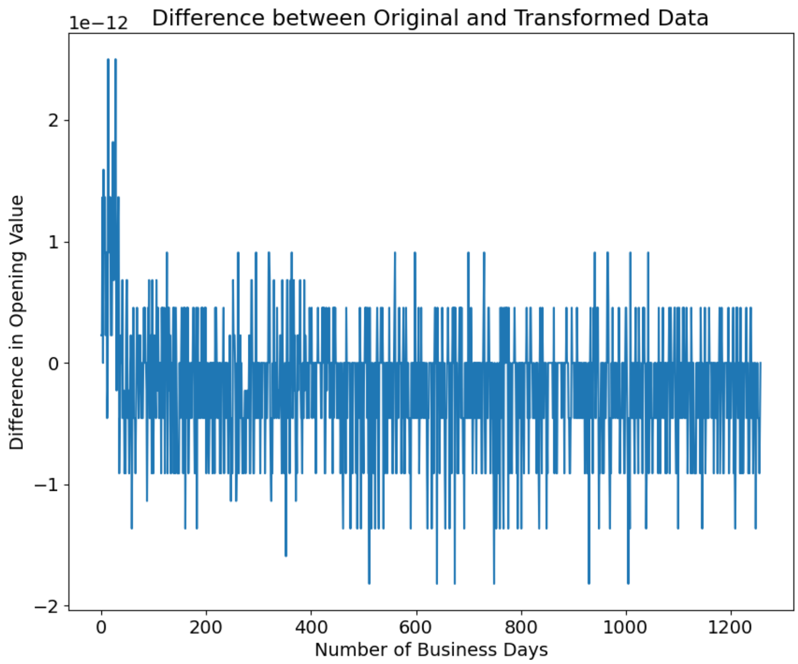

Figure 7: Difference of original S&P 500 data from its Fourier transformed and then inverse Fourier transformed data. Errors of the order 10^-12, which to be expected of round-off error for values of order 10^3.

3 Analysis of Sea Level Pressure

The Fourier transform is also useful in atmospheric physics, specifically with respect to Sea Level Pressure (SLP). The Sea Level Pressure indicates low pressure (associated with cloudy conditions and rain) and high pressure centers (associated with fair and sunny conditions) on the globe. Fig. 8 shows the deviation of SLP from the mean, at 50° Latitude, over all longitudes over 120 days of 2015. This data is from NCEP Reanalysis.

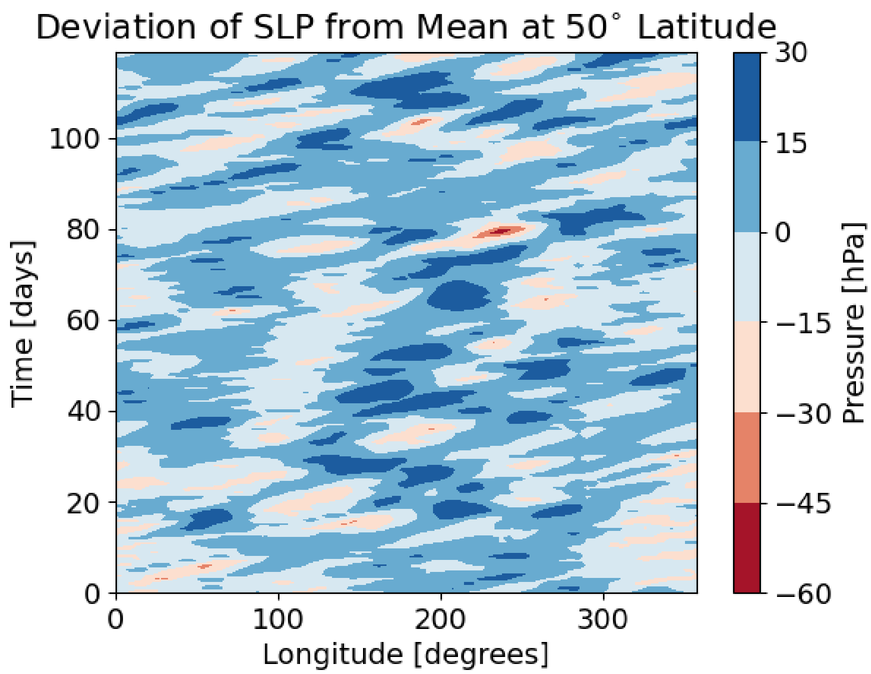

Figure 8: Deviation of Sea Level Pressure from the mean across all longitudes over 120 days, at 50° latitude.

Atmospheric wave propagation theories predict that SLP can be decomposed into a combination of waves with wave number m:

$$SLP(\lambda, t) = \sum_m A(t) cos(m\lambda + \phi(t)) \tag{3}$$

where λ is the longitude, t is time, $A(t)$ and $\phi(t)$ are time-dependent amplitude and phase. We can decompose our data into its Fourier transform along the longitude space and explore the impact of wave number $m = 3$ and $m = 5$. In this analysis, Fourier magnitudes corresponding to all wave numbers other than 3 and 5 were set to zero. The resulting data is seen in Fig. 9.

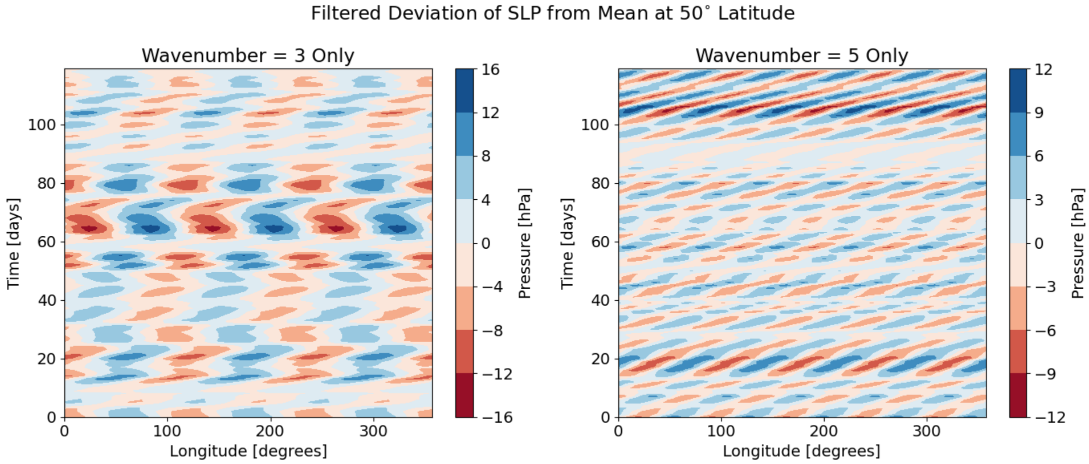

Figure 9: Sea Level Pressure difference extracted at wave numbers 3 and 5 as in Eq.3. Motion of pressure centers toward higher longitudes seen. Higher wave number shows faster motion.

We see that along any time slice, 3 or 5 complete pressure oscillations occur depending on the wave number, as expected from Eq.3. The oscillations along the time space seem much faster for shorter wavelengths ($m = 3$) compared to longer wavelengths ($m = 5$), for which even over a hundred days a full oscillation hasn't completed. The most interesting aspect is the motion of the high or low pressure regions toward higher longitudes (eastward). We see that for shorter wavelengths the pressure regions propagate much faster, about 200° in 20 days for $m = 3$ compared to about 65 days for $m = 5$.

References

[1] Newman, M. E. J. (2013). Computational physics. Revised and expanded 2013.

[2] Harris, Charles R., et al. "Array programming with NumPy." Nature 585.7825 (2020): 357-362. https://doi.org/10.1038/s41586-020-2649-2.