Time Frequency Analysis

Moo K. Chung
University of Wisconsin-Madison
mkchung@wisc.edu

1 Introduction

Time series analysis is a fundamental tool in understanding data that evolves over time. It encompasses various techniques for analyzing **sequential data**, identifying patterns, and making predictions. Traditional statistical methods such as autoregressive models, moving averages, and spectral analysis have been widely used to examine trends and periodicity in time-dependent data. However, many real-world signals, such as biomedical and financial data, exhibit **non-stationary** behavior, where their statistical properties (such as variance) change over time. This necessitates advanced analytical techniques that can capture time-dependent variations effectively. A highly effective data-driven approach, requiring no explicit model specification, is **time-frequency analysis**. This method enables the examination of signal characteristics simultaneously in both time and frequency domains in a multi-scale fashion. This is the focus of this lecture.

2 Fourier Analysis

Traditional **Fourier analysis** provides a global frequency representation but lacks temporal resolution. Given a time series $x(t)$, **Fourier transform** decomposes the signal into a sum of sinusoidal components:

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt, \tag{1}$$

where $e^{-j\omega t}$ is a basis, where the frequency components from the signal is extracted. $X(\omega)$ represents the frequency-domain representation of $x(t)$. $X(\omega)$ is the projection of signal onto basis $e^{-j\omega t}$. ω denotes angular frequency. The **magnitude** $|X(\omega)|$ determines the strength (or amplitude) of each frequency component, and measures how much of each frequency ω is present in the signal. The **phase** $\theta(\omega) = \arg(X(\omega))$ determines the phase shift of each frequency component.

The **inverse Fourier transform** reconstructs the original time-domain signal $x(t)$ from its frequency-domain representation $X(\omega)$ as follows

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{j\omega t} d\omega. \tag{2}$$

This integral expresses the time-domain signal as a sum (integral) of infinitely many complex exponentials of different frequencies ω. Thus, the inverse Fourier transform effectively reconstructs $x(t)$ by combining all frequency components $e^{j\omega t}$, each scaled by its corresponding magnitude and phase from $X(\omega)$.

3 Numerical Implementation

In practical applications, we often work with discrete-time signals, where the continuous Fourier transform is replaced by the Discrete Fourier Transform (DFT). The DFT converts a sequence of N time-domain samples $x[n]$ into its frequency-domain representation $X[k]$ as

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}, \quad k = 0, 1, \ldots, N-1. \tag{3}$$

The DFT provides a numerical approximation of the Fourier transform for sampled signals and is widely used in digital signal processing, spectral analysis, and filtering applications.

To reconstruct a discrete-time signal from its frequency components, we use the inverse discrete Fourier transform (IDFT), given by

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi}{N}kn}, \quad n = 0, 1, \ldots, N-1. \tag{4}$$

This operation reconstructs the original time-domain sequence from its spectral representation. The IDFT is numerically implemented using the inverse Fast Fourier Transform (IFFT), which provides an efficient way to compute the IDFT. The DFT and IDFT are commonly computed using built-in functions in numerical computing environments such as MATLAB and Python. In MATLAB, the functions `fft` and `ifft` are used to compute the DFT and IDFT, respectively.

The direct computation of the DFT and IDFT using the above formule requires $O(N^2)$ operations, making it computationally expensive for large N. This complexity arises because, for each of the N frequency components, we must compute a summation over N time-domain samples, leading to a total of $N \times N = N^2$ operations. To overcome this limitation, the Fast Fourier Transform (FFT) is employed, which reduces the complexity to $O(N \log N)$, making it significantly faster.

While Fourier analysis effectively identifies the dominant frequency components of a signal, it does not provide information about when these frequencies occur. This limitation makes it unsuitable for analyzing non-stationary

signals, where the frequency content varies over time. To address this, time-frequency analysis methods, such as the Short-Time Fourier Transform (STFT) and wavelet transform, introduce localized frequency representations that capture temporal variations.

4 Spectrogram

The **spectrogram**, computed using the **Short-Time Fourier Transform (STFT)**, overcomes this limitation by computing localized frequency content over time. By dividing the signal into overlapping segments and computing the Fourier transform within each segment, the STFT enables visualization of frequency variations over time.

Given a time series signal $x(t)$, the STFT is defined as:

$$X(\tau, \omega) = \int_{-\infty}^{\infty} x(t) w(t-\tau) e^{-j\omega t} dt, \tag{5}$$

where $w(t-\tau)$ is a window function centered at τ that controls the time resolution. ω is the angular frequency. $X(\tau, \omega)$ is the time-frequency representation of $x(t)$. The construct is similar to the wavelet transform, which also provides a time-frequency representation but with adaptive resolution, allowing for better localization of transient features. A common choice for the window function is the Gaussian window

$$w(t) = e^{-\frac{t^2}{2\sigma^2}}, \tag{6}$$

where σ determines the trade-off between time and frequency resolution.

The **power spectrum** or **power spectral density** is the squared magnitude of the STFT

$$S(\tau, \omega) = |X(\tau, \omega)|^2. \tag{7}$$

The power spectrum measures how the energy of a signal is distributed across different frequency components and time (within a given window). It quantifies the strength of each frequency present in the signal, providing insight into its dominant oscillatory behavior. This representation is particularly useful for distinguishing periodic patterns, identifying key frequency components, and analyzing signal characteristics over time. *In stationary signals, the power spectrum remains constant, whereas in non-stationary signals, it evolves over time, requiring time-frequency methods such as spectrogram analysis.* Applications of the power spectrum span various fields, including EEG and fMRI analysis in neuroscience, phoneme detection in speech processing, and vibration analysis in mechanical systems.

The discretization is done by segmenting the signal into overlapping windows, each of which is analyzed separately in the time-frequency domain. The window is typically shifted by an overlap factor, such as 50% of the window length, to ensure smooth transitions between segments and to retain temporal continuity. STFT is then computed for each windowed segment, producing a localized frequency representation. Finally, the spectrogram is obtained by

plotting the power spectrum $S(\tau, \omega)$ as a function of time τ and frequency ω, providing a detailed visualization of how the signal's frequency content evolves over time.