Mathematische Methoden

Contents

1 Fourier Transforms

Analysis - Converts from time-domain to frequency domain

Synthesis - Synthesises the approximated frequency-domain signal back to the time-domain by superposing its sinusoids.

The goal is to find a set of periodic functions, such that their superposition matches the signal being analysed. This is usually done by finding the suitable sinusoid for each possible frequency, changing the time-domain signal to frequency-domain sets of amplitudes and phase shifts.

To be able to capture an arbitrary signal in the frequency domain, we need to determine the amplitude, frequency and phase difference of each possible frequency. A possible basis could be a single sin function:

$$\rho(\omega)\sin(\omega t + \varphi(\omega))$$

Where $\rho(\omega)$ is the amplitude distribution and $\varphi(\omega)$ the phase shift distribution for all frequencies which we must determine. There are variants such as DCT and wavelet decomposition which work in this way. In fact, there are endless possible bases which can capture any signal, such bases are called **complete**.

However unless there is a compelling reason to use an alternative basis, Fourier transforms are done using complex numbers as polar coordinates in the complex plane. This representation has many advantages making them suitable:

- We can represent them in a compact, easy to manipulate way using Euler's formula:

$$Re^{i\omega t} = R(\cos(\omega t) + i\sin(\omega t))$$

This is especially useful when solving for the Fourier transform of a known integrable function (rather than sampling a signal digitally), as powers of e are easy to combine and simplify.

- They encode amplitude and phase for a given frequency, where the amplitude is simply the radius in the complex plane and the angle is the initial phase shift.

When considering sin and cos specifically:

- They are infinitely differentiable
- When considering the space of continuous functions with inner product $< f, g >= \int_a^b f(t)\overline{g(t)}dt$, they are an orthogonal basis:

Orthogonal:

$$< \sin, \cos >= \int_0^{2\pi} \sin(x)\cos(x)dx = \sin^2(x)\ |_0^{2\pi}$$

$$= 0$$

Linearly Independent:

$$\alpha\sin(x) + \beta\cos(x) = 0 \forall x \in [0, 2\pi] \Leftrightarrow \alpha = \beta = 0$$

TODO: Show that they span the entire space.

Here is the summary of the most popular forms of continuous Fourier transform analysis / synthesis functions:

Summary of popular forms of the Fourier transform, one-dimensional

ordinary frequency ξ (Hz)	unitary	$\widehat{f_1}(\xi) \triangleq \int_{-\infty}^{\infty} f(x)\,e^{-i2\pi\xi x}\,dx = \sqrt{2\pi}\,\widehat{f_2}(2\pi\xi) = \widehat{f_3}(2\pi\xi)$ $f(x) = \int_{-\infty}^{\infty} \widehat{f_1}(\xi)\,e^{i2\pi x\xi}\,d\xi$
angular frequency ω (rad/s)	unitary	$\widehat{f_2}(\omega) \triangleq \dfrac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x)\,e^{-i\omega x}\,dx = \dfrac{1}{\sqrt{2\pi}}\,\widehat{f_1}\!\left(\dfrac{\omega}{2\pi}\right) = \dfrac{1}{\sqrt{2\pi}}\,\widehat{f_3}(\omega)$ $f(x) = \dfrac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \widehat{f_2}(\omega)\,e^{i\omega x}\,d\omega$
	non-unitary	$\widehat{f_3}(\omega) \triangleq \int_{-\infty}^{\infty} f(x)\,e^{-i\omega x}\,dx = \widehat{f_1}\!\left(\dfrac{\omega}{2\pi}\right) = \sqrt{2\pi}\,\widehat{f_2}(\omega)$ $f(x) = \dfrac{1}{2\pi} \int_{-\infty}^{\infty} \widehat{f_3}(\omega)\,e^{i\omega x}\,d\omega$

1.1 Negative Frequencies

These can give the notion of sinusoids progressing in the negative direction ie clockwise on the unit circle. However, in many Fourier transforms they do not play a huge role and don't appear as part of the frequency spectrum. Nonetheless, it is necessary to include these frequencies in the analysis to be able to characterize certain complex valued signals.

1.2 Analysis (Transform)

The goal here is to convert a time-domain signal $f(t)$ to a frequency $\rightarrow$ complex number (amplitude and initial phase encoded) mapping, usually denoted as $\hat{f}(\omega)$. This can then be used to analyze the presence of specific frequencies / phase shifts in the signal (and when needed to modify the signal, for example removing parasitic frequencies and resynthesize the signal).

I will consider the angular frequency version of this transform as it feels the most intuitive to me:

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t}dt$$

This results in the fourier transform $\hat{f}(\omega)$ with a complex range. When we plot the radius against frequency graph for all of these complex outputs, we arrive at the top left graph, which can be extremely useful for identifying interesting or unwanted frequencies which are part of the signal:
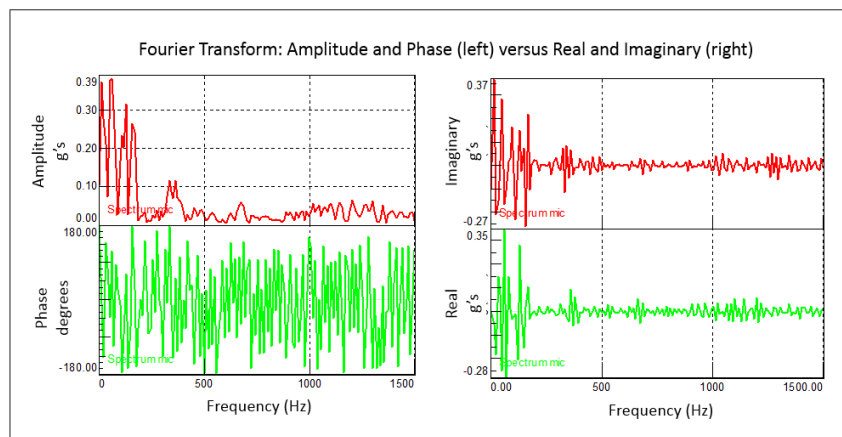
Since the phase is usually not that interesting, it is often discarded unless we plan on synthesizing the time domain signal again.

1.2.1 How It Works - Dot Products

The analysis mechanism can be thought of as a dot product between the signal function $f(t)$ and Fourier basis $e^{i\omega t}$ (or a different basis in other transforms) at a given frequency - a measure of how much the basis at this frequency and phase shift "points in the same direction" as the signal.

Coherency - Constant phase difference due to the same frequency.

To simplify the process, consider the inner product of two cosine functions at frequency ω_1 and ω_2, with phase shifts θ and φ:

$$f := \cos(\omega_1 t + \theta), g := \cos(\omega_2 t + \varphi)$$

$$< f, g >= \int_a^b f(t)\overline{g(t)}dt$$

$$= \int_{-\infty}^{\infty} \cos(\omega_1 t + \theta)\cos(\omega_2 t + \varphi)dt$$

Desmos: https://www.desmos.com/calculator/q8lihjtsur

Their inner product will clearly be at its maximum when the frequencies are equal (they are coherent) and they are in phase; when this is the case they are positive at the same and also negative at the same time, leading to a positive infinitesimal product at all times and a very large integral. Because the signal is not periodic, we need to calculate this dot product over the entire sample (interval $(-\infty, \infty)$).

Replacing $f(t)$ with the time-domain signal and $g(t)$ with the **complex conjugate** of the unit circle $e^{-i\omega t}$ (which encodes both frequency and phase shift) we arrive at the Fourier transform equation:

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t}dt$$

The angle of the summed complex numbers becomes the phase-shift for the transform at that frequency. The complex plot of this has been visualized very well by 3B1B: https://www.youtube.com/watch?v=spUNpyF58BY However, I found his explanation too beginner-friendly.

1.3 Synthesis (Inverse Transform)

We can also recreate the time domain signal from the complex frequency-domain transform:

$$f(t) = \int_{-\infty}^{\infty} \hat{f}(\omega)e^{i\omega t}d\omega$$

This effectively calculates the superposition of all transformed frequencies at a specific point in time.

As seen previously $\hat{f}(\omega)$ returns a complex number for each frequency in the form $Re(i\theta)$ where R is the amplitude of the sinusoid and θ is that frequencies phase shift. When multiplied with $e^{i\omega t}$:

$$\hat{f}(\omega)e^{i\omega t} = Re^{i\theta}e^{i\omega t}$$

$$= Re^{i(\omega t + \theta)}$$

$$= R\cos(\omega t + \theta) + iR\sin(\omega t + \theta)$$

We arrive at the following familiar $R\cos(\omega t + \theta)$ representation of a sinusoid for that given frequency.

As these complex numbers are summed for all possible frequencies at a point in time (superposition of the transforms), the imaginary components cancel each other out due to the orthogonality of sin

and cos, resulting in a single real output for each point in time (unless the time-domain signal had complex components in its values).

1.4 Discrete Fourier Transform

In our digital world we often work with discrete samples of a continuous signal.

Bin - Frequency intervals indexed by 0 to $N-1$, represented by k. If the signal was sampled at frequency f_S for a total of N samples, the frequency bin k represents frequencies $f \in \left[-\frac{1}{2}\frac{kf}{N}, \frac{1}{2}\frac{kf}{N}\right]$. The higher the sampling rate, the more samples are collected and the smaller the interval of each frequency bin $\Rightarrow$ a better, albeit slower, analysis.

Whilst taking the dot product for a given frequency, we can of course only consider dimensions for which we have a sample and therefore analyze **intervals of frequencies**, hence the Discrete Fourier Transform Consider based on N samples is:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi\frac{k}{N}n}$$

Inverse transform:

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} X_k e^{i2\pi\frac{k}{N}n}$$

Notice the frequency represented as $\frac{k}{N} = \frac{\text{Current bin}}{\text{Number of Samples}} \in [0,1[$ has no real-world meaning other than a sense traversing through the range of possible frequencies in the signal. The sample rate of the time-domain signal is not relevant during the Fourier analysis / synthesis and is reintroduced based on the needs of the application, for example when plotting the frequency spectrum / outputing the resynthesized signal. After all, Fourier analysis is just a change of basis, in this case in a finite dimensional space.

Nyquist Theorem - The maximum frequency that can be accurately captured by sample rate of f_s is $\frac{f_s}{2}$. The middle frequency bin $k = \frac{N-1}{2}$ is therefore called the Nyquist bin, and subsequent bins are mirrors of the previous analysis (see Nyquist limit).

See `../../semester-1/lineare-algebra/notes.pdf` for matrix implementation and the Fast Fourier Transform.

TODO Move somewhere more relevant (Signals and Systems Semester 3?): Cool primer in digital signal processing: https://jackschaedler.github.io/circles-sines-signals/ TODO:
• Nyquist frequency, crossing the limit, wagon wheel effect
• Analogue Spectrum analyzers

1.5 Fourier Series

This is the process of converting a periodic signal against time to the frequency domain.

TODO:
• Uncertainty principle
• Analyse how it is a change of basis in function space, decomposing any vector (any function) into the unit vectors ($e^{i\theta}$) using dot product.

2 Proposed Project: Digital Spectrum Analyzer

Goal: Demonstrate understanding of Fourier Analysis, Linear Algebra and learn C++ fundamentals

Description:

- Implement the discrete Fourier Transform + FFT analysis and synthesis algorithms in C++.
- Input signal as CSV / some standard file format and output Amplitude / Frequency plot + synthesize back into time-domain signal to demonstrate quality of different sample rates.
- Interface into the real world by connecting to oscilloscope or even running on some microcontroller and designing suitable host PCB with ADC etc. (explore area of high-frequency electronics) Most likely only accurate for relatively low frequencies (for example the audible range). It may be a good idea to connect it directly to a microphone and design some kind of audio visualizer.

Tools:

- C++, Blaze or Eigen linear algebra library