6 Fourier and Orthogonality

In this chapter we will explore Fourier series and transforms.

6.1 Fourier Series

Fourier series are mathematical series that can represent a periodic signal as a sum of sinusoids at different amplitudes and frequencies. They are useful for solving for the response of a system to periodic inputs. However, they are probably most important *conceptually*: they are our gateway to thinking of signals in the **frequency domain—**that is, as functions of *frequency* (not time). To represent a function as a Fourier series is to *analyze* it as a sum of sinusoids at different frequencies[1] ω_n and amplitudes a_n. Its **frequency spectrum** is the functional representation of amplitudes a_n versus frequency ω_n.

Let's begin with the definition.

Definition 6.1

The *Fourier analysis* of a periodic function $y(t)$ is, for $n \in \mathbb{N}_0$, period T, and angular frequency $\omega_n = 2\pi n/T$,

$$
\begin{aligned}
a_0 &= \frac{2}{T}\int_T y(t)dt \\
a_n &= \frac{2}{T}\int_T y(t)\cos(\omega_n t)dt \\
b_n &= \frac{2}{T}\int_T y(t)\sin(\omega_n t)dt.
\end{aligned}
$$

1. It's important to note that the symbol ω_n, in this context, is not the natural frequency, but a frequency indexed by integer n.

The *Fourier synthesis* of a periodic function $y(t)$ with analysis components a_n and b_n corresponding to ω_n is

$$y(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(\omega_n t) + b_n \sin(\omega_n t).$$

Let's consider the complex form of the Fourier series, which is equivalent to definition 6.1. It may be helpful to review Euler's formula(s)—see appendix C.4.

Definition 6.2

The *Fourier analysis* of a periodic function $y(t)$ is, for $n \in \mathbb{N}_0$, period T, and angular frequency $\omega_n = 2\pi n/T$,

$$c_{\pm n} = \frac{1}{T} \int_{-T/2}^{T/2} y(t) e^{-j\omega_n t} dt.$$

The *Fourier synthesis* of a periodic function $y(t)$ with analysis components c_n corresponding to ω_n is

$$y(t) = \sum_{n=-\infty}^{\infty} c_n e^{j\omega_n t}.$$

We call the integer n a **harmonic** and the frequency associated with it,

$$\omega_n = 2\pi n/T,$$

the **harmonic frequency**. There is a special name for the first harmonic ($n = 1$): the **fundamental frequency**. It is called this because all other frequency components are integer multiples of it.

It is also possible to convert between the two representations above.

Definition 6.3

The complex Fourier analysis of a periodic function $y(t)$ is, for $n \in \mathbb{N}_0$ and a_n and b_n as defined above,

$$c_{\pm n} = \frac{1}{2} \left(a_{|n|} \mp j b_{|n|} \right)$$

The sinusoidal Fourier analysis of a periodic function $y(t)$ is, for $n \in \mathbb{N}_0$ and c_n as defined above,

$$a_n = c_n + c_{-n} \text{ and}$$

$$b_n = j\left(c_n - c_{-n}\right).$$

The **harmonic amplitude** C_n is

$$\begin{aligned} C_n &= \sqrt{a_n^2 + b_n^2} \\ &= 2\sqrt{c_n c_{-n}}. \end{aligned}$$

A **magnitude line spectrum** is a graph of the harmonic amplitudes as a function of the harmonic frequencies. The **harmonic phase** is

$$\begin{aligned} \theta_n &= -\arctan_2(b_n, a_n) && \text{(see appendix C.2.11)} \\ &= \arctan_2(\Im(c_n), \Re(c_n)). && (6.1) \end{aligned}$$

The illustration of figure 6.1 shows how sinusoidal components sum to represent a square wave. A line spectrum is also shown.

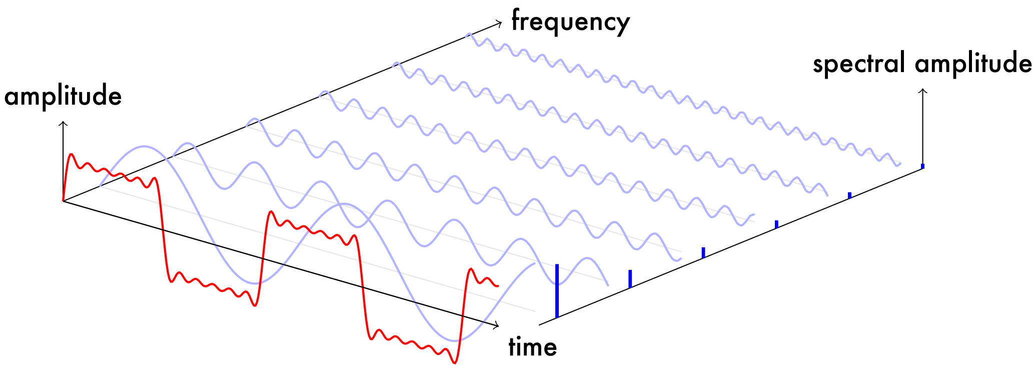

Figure 6.1. A partial sum of Fourier components of a square wave shown through time and frequency. The spectral amplitude shows the amplitude of the corresponding Fourier component.

Let us compute the associated spectral components in the following example.

Example 6.1

Compute the first five harmonic amplitudes that represent the line spectrum for a square wave in the figure above.

Assume a square wave with amplitude 1. Compute a_n:

$$\begin{aligned}a_n &= \frac{2}{T}\int_{-T/2}^{T/2} y(t)\cos(2\pi nt/T)\,dt \\ &= -\frac{2}{T}\int_{-T/2}^{0}\cos(2\pi nt/T)\,dt + \frac{2}{T}\int_{0}^{T/2}\cos(2\pi nt/T)\,dt \\ &= 0 \text{ because cosine is } \textit{even}.\end{aligned}$$

Compute b_n:

$$\begin{aligned}b_n &= \frac{2}{T}\int_{-T/2}^{T/2} y(t)\sin(2\pi nt/T)\,dt \\ &= -\frac{2}{T}\int_{-T/2}^{0}\sin(2\pi nt/T)\,dt + \frac{2}{T}\int_{0}^{T/2}\sin(2\pi nt/T)\,dt \\ &= \frac{2}{n\pi}(1-\cos(n\pi)) \\ &= \begin{cases} 0 & n \text{ even} \\ \frac{4}{n\pi} & n \text{ odd} \end{cases}.\end{aligned}$$

Therefore,

$$\begin{aligned}C_n &= \sqrt{a_n^2 + b_n^2} \\ C_0 &= 0 \text{ (even)} \\ C_1 &= \frac{4}{\pi} \\ C_2 &= 0 \text{ (even)} \\ C_3 &= \frac{4}{3\pi} \\ C_4 &= 0 \text{ (even)} \\ C_5 &= \frac{4}{5\pi}.\end{aligned}$$

6.2 Fourier Transform

0J

We begin with the usual loading of modules.

```
import numpy as np # for numerics
import sympy as sp # for symbolics
import matplotlib.pyplot as plt # for plots!
```

Let's consider a periodic function f with period T (`T`). Each period, the function has a triangular pulse of width δ (`pulse_width`) and height $\delta/2$.

```
period = 15 # period
pulse_width = 2 # pulse width
```

First, we plot the function f in the time domain. Let's begin by defining f.

```
def pulse_train(t,T,pulse_width):
  f = lambda x:pulse_width/2-abs(x) # pulse
  tm = np.mod(t,T)
  if tm <= pulse_width/2:
    return f(tm)
  elif tm >= T-pulse_width/2:
    return f(-(tm-T))
  else:
    return 0
```

Now, we develop a numerical array in time to plot f.

```
N = 151 # number of points to plot
tpp = np.linspace(-period/2,5*period/2,N) # time values
fpp = np.array(np.zeros(tpp.shape))
for i,t_now in enumerate(tpp):
  fpp[i] = pulse_train(t_now,period,pulse_width)
```

Now we plot.

```
fig, ax = plt.subplots()
ax.plot(tpp,fpp,'b-',linewidth=2) # plot
plt.xlabel('time (s)')
plt.xlim([-period/2,3*period/2])
plt.xticks(
  [0,period],
  [0,'$T='+str(period)+'$ s']
)
plt.yticks([0,pulse_width/2],['0','$\delta/2$'])
plt.draw()
```