6.3000: Signal Processing

Sinusoids and Fourier Series

February 06, 2025

Fourier Series (Trigonometric Form)

If $f(t)$ is expressed as a Fourier series

$$f(t) = f(t+T) = c_0 + \sum_{k=1}^{\infty} \Big(c_k \cos(k\omega_o t) + d_k \sin(k\omega_o t) \Big)$$

the Fourier coefficients are given by

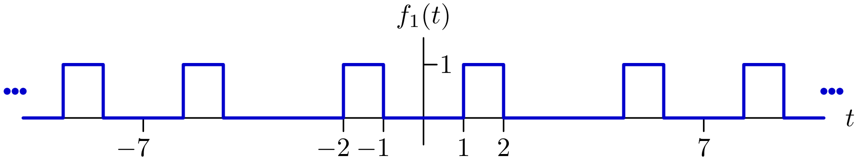
$$c_0 = \frac{1}{T} \int_T f(t)\, dt$$

$$c_k = \frac{2}{T} \int_T f(t) \cos(k\omega_o t)\, dt; \;\; k = 1, 2, 3, \ldots$$

$$d_k = \frac{2}{T} \int_T f(t) \sin(k\omega_o t)\, dt; \;\; k = 1, 2, 3, \ldots$$

Two Pulses

Let $f_1(t)$ represent the following function, which is periodic in $T = 7$:



Determine a Fourier series of the following form for $f_1(t)$.

$$f_1(t) = c_0 + \sum_{k=1}^{\infty} \Big(c_k \cos(k\omega_o t) + d_k \sin(k\omega_o t) \Big)$$

Two Pulses

The average value is $c_0 = \dfrac{1}{T}\displaystyle\int_T f_1(t)\,dt$.

There are many equivalent ways to integrate over a period:

$$\int_T dt = \int_0^7 dt = \int_{-7/2}^{7/2} dt = \cdots$$

All of these are easy. The result is $c_0 = \dfrac{2}{7}$.

Two Pulses

For $k \geq 1$:

$$c_k = \frac{2}{T} \int_T f_1(t) \cos(k\omega_o t)\, dt$$

Integrating over symmetric limits simplifies the math (slightly):

$$\begin{aligned} c_k &= \frac{2}{7} \int_{-2}^{2} \cos(k\omega_o t)\, dt - \frac{2}{7} \int_{-1}^{1} \cos(k\omega_o t)\, dt \\ &= \frac{2}{7} \left. \frac{\sin(k\omega_0 t)}{k\omega_o} \right|_{-2}^{2} - \frac{2}{7} \left. \frac{\sin(k\omega_0 t)}{k\omega_o} \right|_{-1}^{1} \\ &= \frac{2}{\pi k}\left(\sin \frac{4k\pi}{7} - \sin \frac{2k\pi}{7} \right) \end{aligned}$$

Demonstrate other ways to do this integration, e.g.,

$$\int_{-2}^{1} dt + \int_{1}^{2} dt \quad \text{or} \quad \int_{1}^{2} dt + \int_{6}^{7} dt \quad \text{or} \quad 2\int_{1}^{2} dt$$

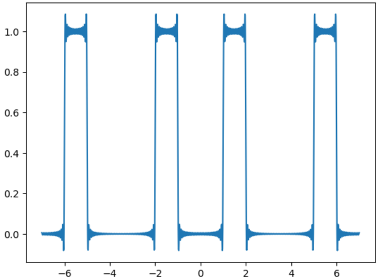
Since $f_1(t)$ is symmetric, and harmonics of the sine function are antisymmetric, the d_k coefficients are all zero.

$$d_k = \frac{2}{T} \int_T f_1(t) \sin(k\omega_o t)\, dt = 0$$

Checking with Python

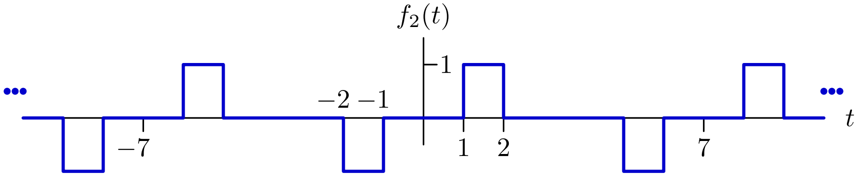
Sum the first 100 elements of series:

```
from math import sin, pi
from matplotlib.pyplot import plot, show
x = []
y = []
omega0 = 2*pi/7
t = -7
while t<7:
    x.append(t)
    y.append(2/7+sum([2*(sin(4*pi*k/7)-sin(2*pi*k/7))/pi/k*cos(k*omega0*t)
             for k in range(1,100)]))
    t += 0.01
plot(x,y)
show()
```



Opposite Pulses

Let $f_2(t)$ represent the following function, which is periodic in $T = 7$:



Find ω_o and the Fourier series coefficients c_k and d_k so that

$$f_2(t) = \sum_{k=0}^{\infty} \Big(c_k \cos(k\omega_o t) + d_k \sin(k\omega_o t) \Big)$$

Opposite Pulses

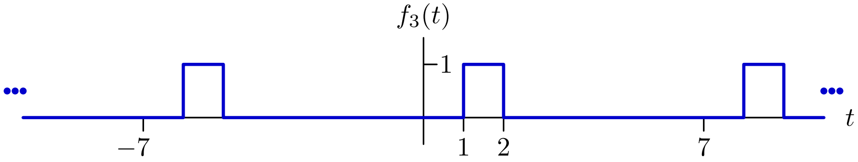
The previous problem was symmetric about $t = 0$ so there were only cosine terms.

This problem is antisymmetric about $t = 0$ so there are only sine terms.

Do the calculation using several different regions of integration.

Single Pulse

Let $f_3(t)$ represent the following function, which is periodic in $T = 7$:



Determine the Fourier series coefficients for $f_3(t)$.

Discuss the relation(s) among the Fourier series coefficients of $f_1(t)$, $f_2(t)$, and $f_3(t)$.

Discuss the relation(s) among $f_1(t)$, $f_2(t)$, and $f_3(t)$.

Check With Python

It's interesting that overshoots move around. They are "glued" to the discontinuities.

Trig Table

```
sin(a+b) = sin(a) cos(b) + cos(a) sin(b)
sin(a-b) = sin(a) cos(b) - cos(a) sin(b)
cos(a+b) = cos(a) cos(b) - sin(a) sin(b)
cos(a-b) = cos(a) cos(b) + sin(a) sin(b)
tan(a+b) = (tan(a)+tan(b))/(1-tan(a) tan(b))
tan(a-b) = (tan(a)-tan(b))/(1+tan(a) tan(b))

sin(A) + sin(B) =  2 sin((A+B)/2) cos((A-B)/2)
sin(A) - sin(B) =  2 cos((A+B)/2) sin((A-B)/2)
cos(A) + cos(B) =  2 cos((A+B)/2) cos((A-B)/2)
cos(A) - cos(B) = -2 sin((A+B)/2) sin((A-B)/2)

sin(a+b) + sin(a-b) =  2 sin(a) cos(b)
sin(a+b) - sin(a-b) =  2 cos(a) sin(b)
cos(a+b) + cos(a-b) =  2 cos(a) cos(b)
cos(a+b) - cos(a-b) = -2 sin(a) sin(b)

2 cos(A) cos(B) = cos(A-B) + cos(A+B)
2 sin(A) sin(B) = cos(A-B) - cos(A+B)
2 sin(A) cos(B) = sin(A+B) + sin(A-B)
2 cos(A) sin(B) = sin(A+B) - sin(A-B)
```