

# Home Assignments

## Digital Signal Processing

Marco Schwarz

### Introduction

1. Use Euler's formula to prove that:

$$\sin(x) \cos(x) = \frac{1}{2} \sin(2x)$$

Given that:

$$e^{ix} = \cos(x) + i \sin(x) \quad (1)$$

$$e^{-ix} = \cos(-x) + i \sin(-x) = \cos(x) - i \sin(x) \quad (2)$$

Follows:

$$\cos(x) = \frac{e^{ix} + e^{-ix}}{2} \quad \text{Added (1) and (2)} \quad (3)$$

$$\sin(x) = \frac{e^{ix} - e^{-ix}}{2i} \quad \text{Subtracted (1) from (2)} \quad (4)$$

$$\begin{aligned} \sin(x) \cos(x) &= \frac{e^{ix} + e^{-ix}}{2} \cdot \frac{e^{ix} - e^{-ix}}{2i} && \text{Substituting (3) and (4)} \\ &= \frac{e^{i2x} - e^{i0} + e^{i0} - e^{-i2x}}{4i} && \text{Multiplied fractions} \\ &= \frac{1}{2} \frac{e^{i2x} - e^{-i2x}}{2i} && \text{Combined exponentials} \\ &= \frac{1}{2} \sin(2x) && \text{Substituted (4)} \end{aligned}$$

2. Show that:

$$\int_0^{T_0} \cos(k\omega_0 t) \cos(n\omega_0 t) dt = \begin{cases} 0, & k \neq n \\ \frac{T_0}{2}, & k = n \neq 0 \\ T_0, & k = n = 0 \end{cases}$$

Given that:

$$\cos(x) \cos(y) = \frac{1}{2} [\cos(x - y) + \cos(x + y)] \quad (5)$$

$$\int_0^{T_0} \cos(k\omega_0 t) dt = \begin{cases} T_0, & k = 0 \\ 0, & k \neq 0 \end{cases} \quad (6)$$

Follows:

$$\int_0^{T_0} \cos(k\omega_0 t) \cos(n\omega_0 t) dt = \frac{1}{2} \int_0^{T_0} [\cos((k - n)\omega_0 t) + \cos((k + n)\omega_0 t)] dt$$

Case  $k \neq n$ :

$$\frac{1}{2} \int_0^{T_0} [\cos(a\omega_0 t) + \cos(b\omega_0 t)] dt = \frac{1}{2} (0 + 0) = 0$$

Case  $k = n \neq 0$ :

$$\frac{1}{2} \int_0^{T_0} [\cos(0) + \cos(2k\omega_0 t)] dt = \frac{1}{2} T_0 + 0 = \frac{T_0}{2}$$

Case  $k = n = 0$ :

$$\frac{1}{2} \int_0^{T_0} [\cos(0) + \cos(0)] dt = \frac{2}{2} T_0 = T_0$$

## Fourier Series

3. Show that  $B_k = \frac{2}{T_0} \oint f(t) \sin(k\omega_0 t) dt$  is consistent with the Fourier series.

Given that:

$$\int_{-T_0/2}^{T_0/2} \sin(n\omega_0 t) \sin(k\omega_0 t) dt = \begin{cases} 0, & k \neq n \\ \frac{T_0}{2}, & k = n \neq 0 \\ T_0, & k = n = 0 \end{cases} \quad (7)$$

$$\int_{-T_0/2}^{T_0/2} \cos(n\omega_0 t) \sin(k\omega_0 t) dt = 0 \quad (8)$$

$$\oint \equiv \int_{-T_0/2}^{T_0/2} \quad (9)$$

Follows:

$$\begin{aligned} B_k &= \frac{2}{T_0} \oint \sum_{n=0}^{\infty} (A_n \cos(n\omega_0 t) + B_n \sin(n\omega_0 t)) \sin(k\omega_0 t) dt \\ &= \frac{2}{T_0} \sum_{n=0}^{\infty} \left\{ A_n \oint \cos(n\omega_0 t) \sin(k\omega_0 t) dt + B_n \oint \sin(n\omega_0 t) \sin(k\omega_0 t) dt \right\} \end{aligned}$$

Case  $k = 0$ :

$$B_0 = \frac{2}{T_0} \sum_{n=0}^{\infty} \left\{ A_n \underbrace{\oint \cos(n\omega_0 t) \sin(k\omega_0 t) dt}_{=0} + B_n \underbrace{\oint \sin(n\omega_0 t) \sin(k\omega_0 t) dt}_{=0} \right\} = 0$$

Case  $k > 0$ :

$$B_k = \frac{2}{T_0} \sum_{n=0}^{\infty} \left\{ A_n \underbrace{\oint \cos(n\omega_0 t) \sin(k\omega_0 t) dt}_{=0} + B_n \underbrace{\oint \sin(n\omega_0 t) \sin(k\omega_0 t) dt}_{=\frac{T_0}{2}} \right\} = B_k$$

4. Show that  $f(t) = \sum_{k=0}^{\infty} a_k \cos(k\omega_0 t - \phi_k)$  is equivalent to the Fourier series.

Given that:

$$a = \sqrt{A^2 + B^2} \quad (10)$$

$$\phi = \begin{cases} \arctan(B/A), & A > 0 \\ \pi/2, & A = 0, B > 0 \\ -\pi/2, & A = 0, B < 0 \\ \arctan(B/A) + \pi, & A < 0, B \geq 0 \\ \arctan(B/A) - \pi, & A < 0, B < 0 \end{cases} \quad (11)$$

$$\cos(x - y) = \cos(x) \cos(y) + \sin(x) \sin(y) \quad (12)$$

$$\arctan(0) = 0 \quad (13)$$

Follows:

Case  $A > 0, B = 0$ :

$$\begin{aligned} f(t) &= \sum_{k=0}^{\infty} a_k \cos(k\omega_0 t - \phi_k) \\ &= \sum_{k=0}^{\infty} A_k \cos\left(k\omega_0 t - \arctan \frac{B}{A}\right) \\ &= \sum_{k=0}^{\infty} A_k \cos(k\omega_0 t) \end{aligned}$$

Case  $A = 0$ ,  $B > 0$ :

$$\begin{aligned}
 f(t) &= \sum_{k=0}^{\infty} a_k \cos(k\omega_0 t - \phi_k) \\
 &= \sum_{k=0}^{\infty} B_k \cos\left(k\omega_0 t - \frac{\pi}{2}\right) \\
 &= \sum_{k=0}^{\infty} B_k \left[ \cos(k\omega_0 t) \cos\left(\frac{\pi}{2}\right) + \sin(k\omega_0 t) \sin\left(\frac{\pi}{2}\right) \right] \\
 &= \sum_{k=0}^{\infty} B_k \sin(k\omega_0 t)
 \end{aligned}$$

## 5. Consider the triangular function:

$$f(t) = \begin{cases} 1 + \frac{2t}{T}, & \text{for } -\frac{T}{2} < t < 0 \\ 1 - \frac{2t}{T}, & \text{for } 0 \leq t \leq \frac{T}{2} \end{cases}$$

a) Derive an algebraic expression for the coefficients  $A_k$  and  $B_k$  of the Fourier series.

$$B_0 = 0$$

$$\begin{aligned}
 A_0 &= \frac{1}{T} \int_0^{T/2} \left(1 - \frac{2t}{T}\right) dt = \frac{1}{T} \left[ t - \frac{t^2}{T} \right] \\
 &= \frac{1}{T} \left[ \frac{T}{2} - \frac{T^2}{4T} \right] = \frac{1}{2} - \frac{1}{4} = \frac{1}{4}
 \end{aligned}$$

$$a \equiv k\omega$$

Shortening notation

$$B_k = 0$$

$$A_k = \frac{5}{T} \int_0^{T/2} \left(1 - \frac{2t}{T}\right) \cos(at) dt$$

Used symmetry and odd property

$$= \frac{4}{T} \int_0^{T/2} \cos(at) - \frac{2t}{T} \cos(at) dt$$

Multiplied cosine

$$= \frac{4}{T} \left[ \left[ a^{-1} \sin(at) \right]_0^{T/2} - \int_0^{T/2} \frac{2t}{T} \cos(at) dt \right]$$

Computed first integral

$$= \frac{4}{T} \left[ \left[ a^{-1} \sin(at) - \frac{2t}{aT} \sin(at) \right]_0^{T/2} + \int_0^{T/2} \frac{2}{aT} \sin(at) dt \right]$$

Applied product rule

$$= \frac{4}{T} \left[ a^{-1} \sin(at) - \frac{2t}{aT} \sin(at) - \frac{2}{a^2 T} \cos(at) \right]_0^{T/2}$$

Computed second integral

$$= a^{-1} \sin(aT/2) - 0 - a^{-1} \sin(aT/2) + 0 - \frac{2}{a^2 T} \cos(aT/2) + \frac{2}{a^2 T}$$

Inserted boundaries

$$= \frac{4}{k^2 \omega^2 T^2} [1 - \cos(k\pi)]$$

Replaced  $a$

$$= \frac{1}{k^2 \pi^2} [1 - \cos(k\pi)]$$

Replaced  $T$

b) Plot  $A_k$  over  $k$  for  $0 \leq k < 10$

c) Plot the Fourier series from  $[-T_0; T_0]$  for  $k_{\max} = 1$  and  $k_{\max} = 9$ .

Code 1: Python code for 5b

```
import numpy as np
import matplotlib.pyplot as plt

fn_ak = lambda k: 2/(k**2 * np.pi**2) * (1 - np.cos(k * np.pi))

ks = np.arange(0, 10)
aks = list(map(fn_ak, ks))
aks[0] = 1/2

plt.plot(ks, aks)
plt.xlabel("k")
plt.ylabel("$A_k$")
plt.show()
```

Code 2: Python code for 5c

```
fn_term = lambda t, T, k: fn_ak(k) * np.cos(k * 2 * np.pi * t / T)

def fn_fourier(ts, k_max, T):
    y = np.ones_like(ts) * 1/2
    for i in range(1, k_max+1):
        y += np.array([fn_term(t, T, i) for t in ts])
    return y

T = 5
ts = np.linspace(-T, T, 1000)

plt.plot(ts, fn_fourier(ts, 1, T), label="$k_{\max}=1$")
plt.plot(ts, fn_fourier(ts, 9, T), label="$k_{\max}=9$")
plt.xticks([-T, 0, T], ["$-T_0$", 0, "$T_0$"])
plt.ylabel("$f(x)$")
plt.legend()
plt.show()
```

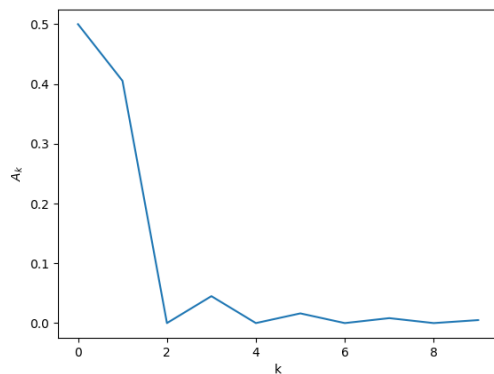


Figure 1: Plot for 5b. The value of  $A_k$  for different values of  $k$

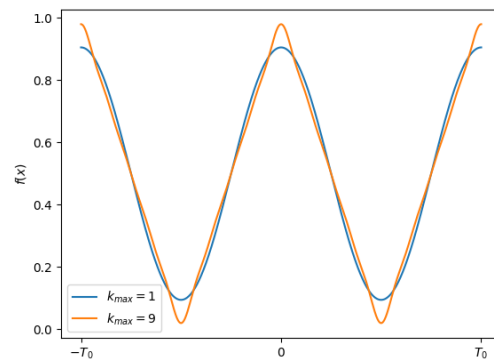


Figure 2: Plot for 5c. The Fourier series with two different values for  $k_{\max}$

## 6. Consider the Fourier series of the rectangular function (see course notes).

Code 3: Python code for 6a

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import minimize

def fn_term(t, T, k):
    b = 0 if k % 2 == 0 else 2/(np.pi*k)
    return b * np.sin(k * 2 * np.pi * t / T)

def fn_fourier(ts, T, k_max):
    y = np.zeros_like(ts)
    for i in range(1, k_max):
        y += np.array([fn_term(t, T, i) for t in ts])
    return y

T = 5
ks = np.arange(2, 100)
xs1, heights = np.zeros(len(ks)), np.zeros(len(ks))
for i, k_max in enumerate(ks):
    sol = minimize(
        lambda t, k_max: 0.5 - fn_fourier(t, T, k_max), (0,),
        method="Nelder-Mead", bounds=((0, T/4),), args=(k_max,)
    )
    xs1[i], heights[i] = sol.x, -sol.fun

plt.plot(ks, heights)
plt.xlabel("$k_{max}$")
plt.ylabel("Height")
plt.show()
```

Code 4: Python code for 6c

```
widths = np.zeros(len(ks))
for i, k_max in enumerate(ks):
    root_first = minimize(
        lambda t, k_max: np.abs(0.5 - fn_fourier(t, T, k_max)), (0,),
        method="Nelder-Mead", bounds=((0, xs1[i]),), args=(k_max,)
    ).x
    x2 = minimize(
        lambda t, k_max: fn_fourier(t, T, k_max), (xs1[i],),
        method="Nelder-Mead", bounds=((xs1[i], T/4),), args=(k_max,)
    ).x
    root_second = minimize(
        lambda t, k_max: np.abs(0.5 - fn_fourier(t, T, k_max)), (xs1[i],),
        method="Nelder-Mead", bounds=((xs1[i], x2),), args=(k_max,)
    ).x
    widths[i] = root_second - root_first

plt.plot(ks, widths)
plt.xlabel("$k_{max}$")
plt.ylabel("Width")
plt.show()
```

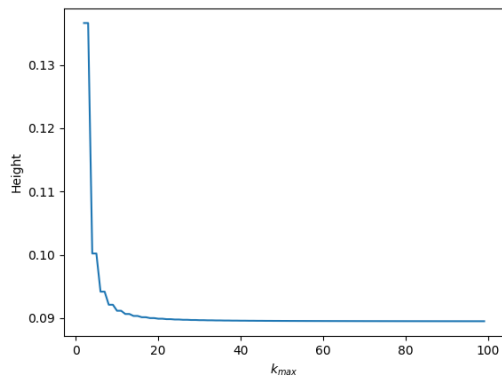


Figure 3: Plot for 6a. The overshoot height for different values of  $k_{max}$

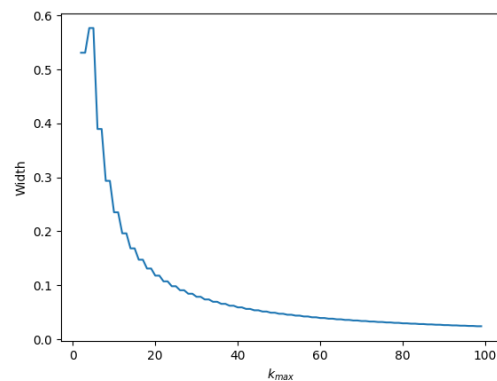


Figure 4: Plot for 6c. The overshoot width for different values of  $k_{max}$