

# РОБОТА із ЗОБРАЖЕННЯМИ

Файл: Image\_07\_001

## Ряд Фур'є

Використовуємо бібліотеку [SCIPI](#)

```
from scipy.fft import fft # модуль для перетворення Фур'є
from scipy.integrate import quad # модуль для інтегрування
import matplotlib.pyplot as plt
import numpy as np
```

## Синтез прямокутних імпульсів

```
def signal(t): # визначає функцію як суму гармонік
    # Постійна складова
    A0 = 5
    # Період сигналу
    T = 1.0
    s = A0/2

    # Перша гармоніка
    A1 = 7; T1 = T; w1=2*np.pi/T1;    Fi1 = 0.0
    s += A1*np.cos(w1*t+Fi1)
    # Третя гармоніка
    A3 = -3; T3 = T/3; w3=2*np.pi/T3;    Fi3 = 0.0
    s += A3*np.cos(w3*t+Fi3)
    # П'ята гармоніка
    A5 = 2; T5 = T/5; w5=2*np.pi/T5;    Fi5 = 0.0
    s += A5*np.cos(w5*t+Fi5)
    # Сьома гармоніка
    A7 = -.5; T7 = T/7; w7=2*np.pi/T7;    Fi7 = 0.0
    s += A7*np.cos(w7*t+Fi7)
    # Дев'ята гармоніка
    A9 = -.1; T9 = T/9; w9=2*np.pi/T9;    Fi9 = 0.0
    s += A9*np.cos(w9*t+Fi9)

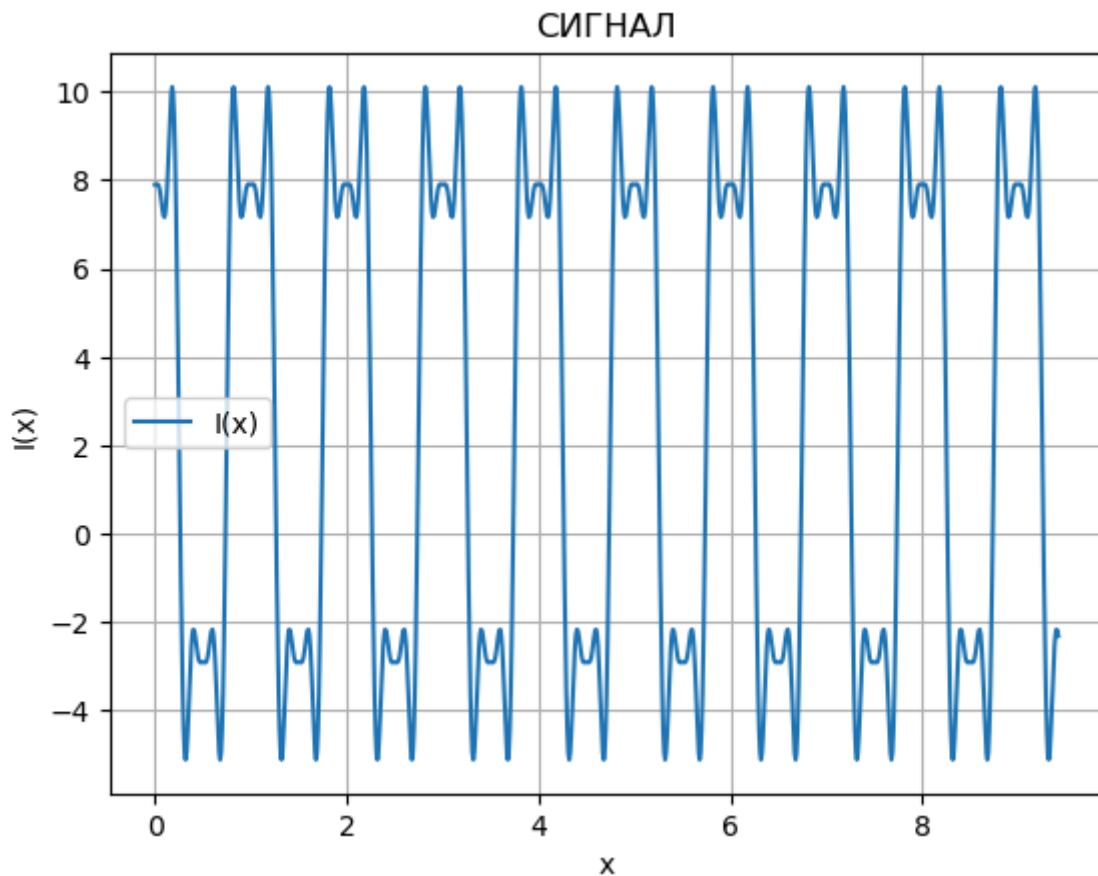
    return s # повертає сигнал
```

```

q=np.arange(0, 3*np.pi, 0.01) # підготовка аргументу функції

plt.figure()
P=[Signal(t) for t in q]
plt.title("СИГНАЛ")
plt.plot(q, P, label='I(x)')
plt.xlabel("x")
plt.ylabel("I(x)")
plt.legend(loc='best')
plt.grid(True)
plt.show()

```



## Прямокутний періодичний сигнал

```

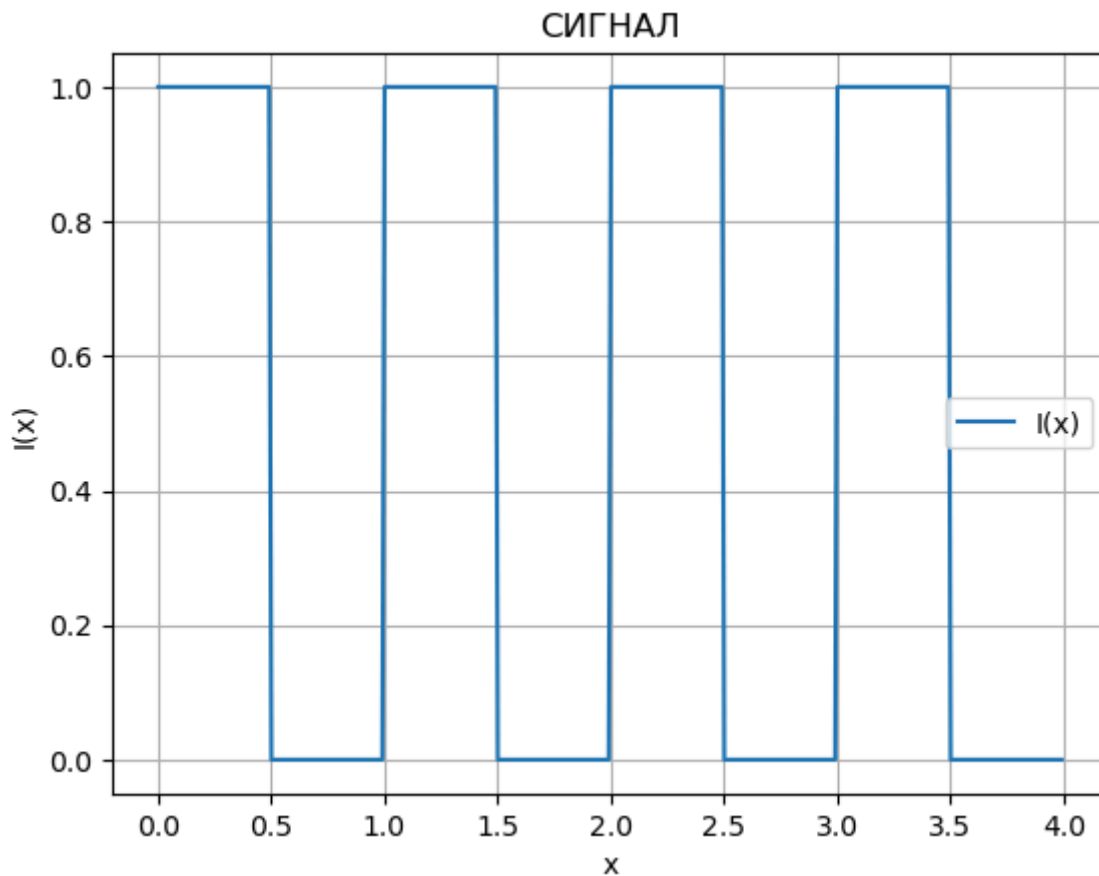
T=1                # період
w=2*np.pi/T       # кругова частота
sq = 2             # скважність

def func(t): # функція сигналу
    if t%T< T/sq:
        p=1
    else:
        p=0
    return p

```

```
q=np.arange(0,4*T,0.01)# підготовка аргумента X для численного анализа
```

```
plt.figure()
P=[func(t) for t in q]
plt.title("СИГНАЛ")
plt.plot(q, P, label='I(x)')
plt.xlabel("x")
plt.ylabel("I(x)")
plt.legend(loc='best')
plt.grid(True)
plt.show()
```



## Обчислення амплітуд гармонік

для відображення змінюємо скважність та кількість гармонік

```
def func_1(t,k,w):# функція для расчёта коэффициента a[k]
    if t%T< T/sq:
        z=func(t)*np.cos(w*k*t)
    else:
        z=-func(t)*np.cos(w*k*t)
    return z

def func_2(t,k,w):#функция для расчёта коэффициента b[k]
    if t%T< T/sq:
        y=func(t)*np.sin(w*k*t)
    else:
        y=-func(t)*np.sin(w*k*t)
```

```
y=func(t)*np.sin(w*k*t)
return y
```

```
# Для відображення задаємо параметр c - кількість гармонік

a=[]; b=[]; g=[]

c=6          # кількість гармонік (на 1 більше)

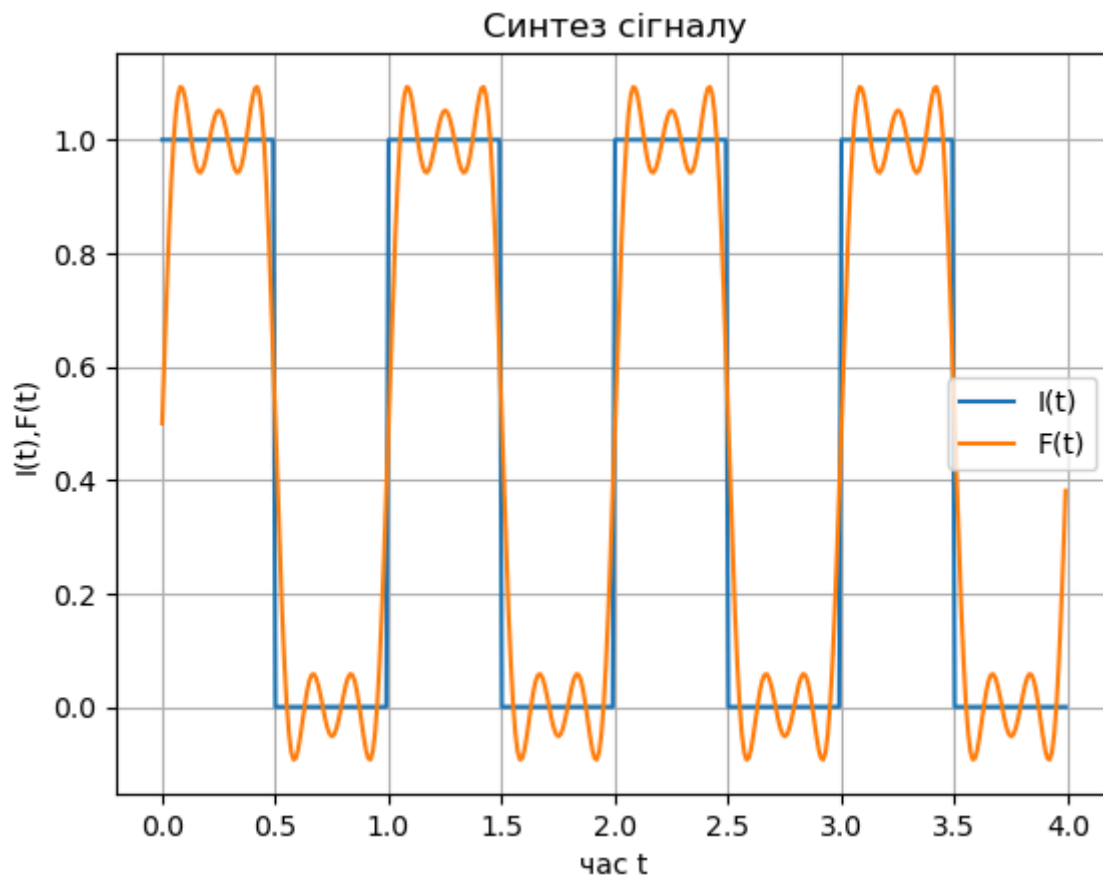
m=np.arange(0,c,1) # індекс гармоніки від 0 до 10-x

a=[round(2*quad(func_1, 0, T, args=(k,w))[0]/T,3) for k in m] # інтеграл для a[k],
k -номер гармоніки
b=[round(2*quad(func_2, 0, T, args=(k,w))[0]/T,3) for k in m] # інтеграл для b[k],
k -номер гармоніки

F=np.array(a[0]/2)+np.array([0*t for t in q-1])# підготовка масиву для аналізу c
a[0]/2
for k in np.arange(1,c,1):
    F=F+np.array([a[k]*np.cos(w*k*t)+b[k]*np.sin(w*k*t) for t in q]) #
обчислення членів ряду Фур'є

P=[func(t) for t in q]
```

```
plt.figure()
plt.title("Синтез сигналу")
plt.plot(q, P, label='I(t)')
plt.plot(q, F, label='F(t)')
plt.xlabel("час t")
plt.ylabel("I(t),F(t)")
plt.legend(loc='best')
plt.grid(True)
plt.show()
```



## Спектр прямокутних імпульсів

```
# Number of sample points
t_max = 2.0    # full interval in seconds
N = 400        # Number of sample points
S = t_max / N  # sample spacing
f_main = 2.0   # ЧАСТОТА ОСНОВНОГО СИГНАЛА  ГЕРЦ
tau_main = 1.0/f_main # ПЕРИОД ОСНОВНОГО СИГНАЛА  сек
print ('Інтервал ', t_max, 'секунд. Кількість відліків', N, 'Довжина відліку', S)
print ('Основна частота сигналу', f_main, 'Hz, Основний період', tau_main )

t = np.linspace(0.0, t_max, N) # start, stop, num

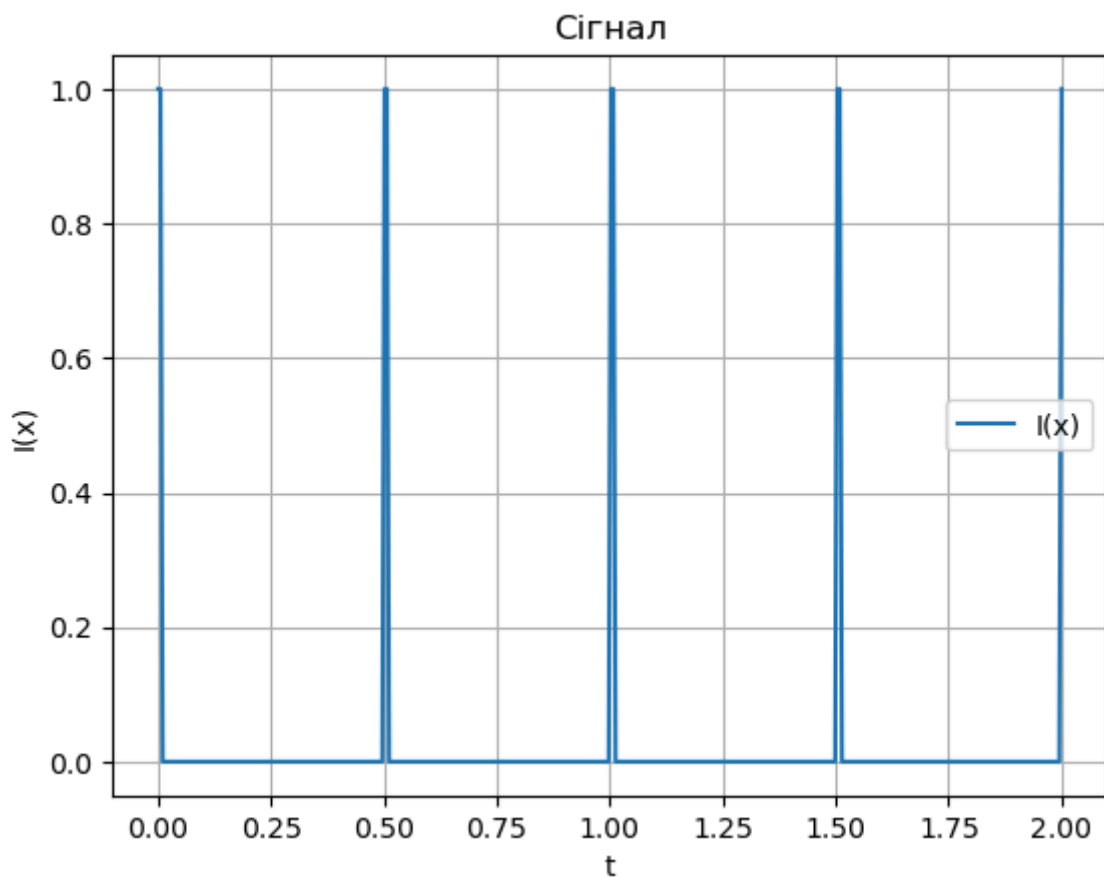
# Вузкі прямокутні імпульси
# Змінюємо скважність

sq = 50 # скважність

def func(x, tau_main): # функція сигналу
    if x%tau_main < tau_main / sq :
        p=1
    else:
        p=0
    return p
y = [func(x, tau_main) for x in t]
```

Інтервал 2.0 секунд. Кількість відліків 400 Довжина відліку 0.005  
Основна частота сигналу 2.0 Hz, основний період 0.5

```
# Сигнал
plt.title("Сигнал")
plt.plot(t, y, label = 'I(x)')
plt.xlabel("t")
plt.ylabel("I(x)")
plt.legend(loc='best')
plt.grid()
plt.show()
```



```
# Швидке перетворення Фур'є
yf = fft(y)
plt.title("Спектр")
xf = np.linspace(0.0, N/(2.0*t_max), N//2)
plt.plot(xf, 2.0/N * np.abs(yf[0:N//2]))
plt.xlabel("Hz")
plt.ylabel("S(f)")
plt.grid()
plt.show()
```

