

1. Proszę wybrać z tabeli 1 funkcję $x(t)$, wygenerować próbki do bufora oraz wykreślić postać uzyskanego sygnału. Należy dokonać samodzielnego wyboru parametrów f , ϕ oraz $f_s \geq 8\text{kHz}$, $T_c \geq 1\text{s}$.
2. Dla dowolnego zestawu funkcji z tabeli 2 należy wygenerować trzy sygnały reprezentujące funkcje $y(t)$, $z(t)$ oraz $v(t)$, gdzie $x(t)$ jest funkcją wybraną w ćwiczeniu 1. Wykonać wykresy dla każdego z wygenerowanych sygnałów przy takich samych parametrach f_s oraz T_c jak w poprzednim ćwiczeniu.
3. Wykreślić wykres dla dowolnej funkcji $u(t)$ wybranej z tabeli 3. Czas trwania sygnału wynika z definicji funkcji, natomiast częstotliwość próbkowania f_s jest taka sama jak w poprzednim ćwiczeniu.
4. Wygenerować i wykreślić sygnały $b_k(t)$ ($k = 1, 2, 3$) dla $f_s = 22.05\text{kHz}$ oraz $T_c = 1\text{s}$.

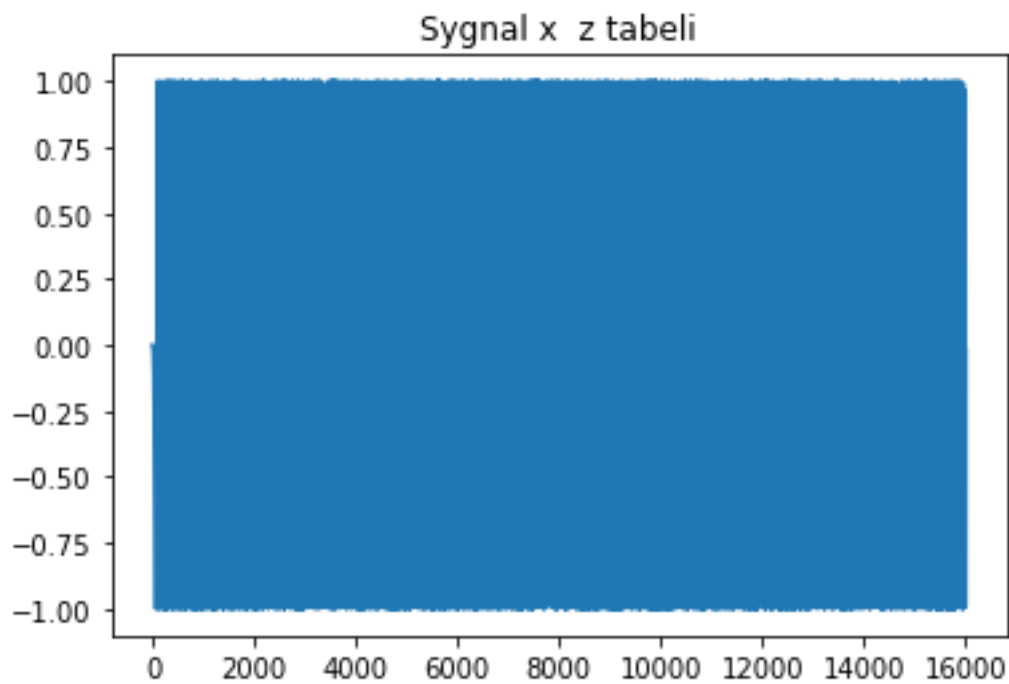
```
import matplotlib.pyplot as plt
import math
from math import sin, cos, pi, sqrt, log
#czestotliwosc probkowania
f_s = 8000

#przesuniecie w fazie
phi = pi/120
#funkcja z tabeli
funkcja_x = lambda t: sin(2*pi*f_s*t*cos(3*pi*t) + t*phi)

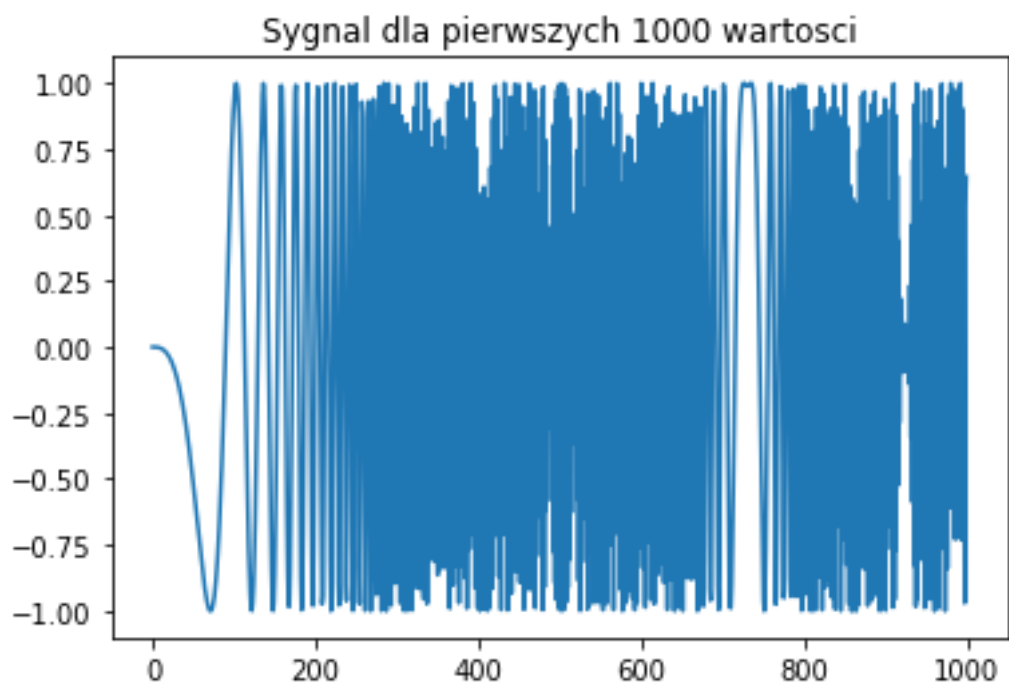
#sekundy
T = 2

def generate_signal(funkcja, f_s, T):
    ''' Funkcja generujaca sygnal dla funkcji: funkcja, czestotliwosci: f_s i czasu trwania T'''
    return [funkcja(n/f_s) for n in range(int(T*f_s))]

#wygenerowany sygnal
x = generate_signal(funkcja_x, f_s, T)
plt.plot(range(T*f_s), x)
plt.title("Sygnal x z tabeli")
```



```
plt.plot(range(1000), x[:1000])
plt.title("Sygnal dla pierwszych 1000 wartosci")
```



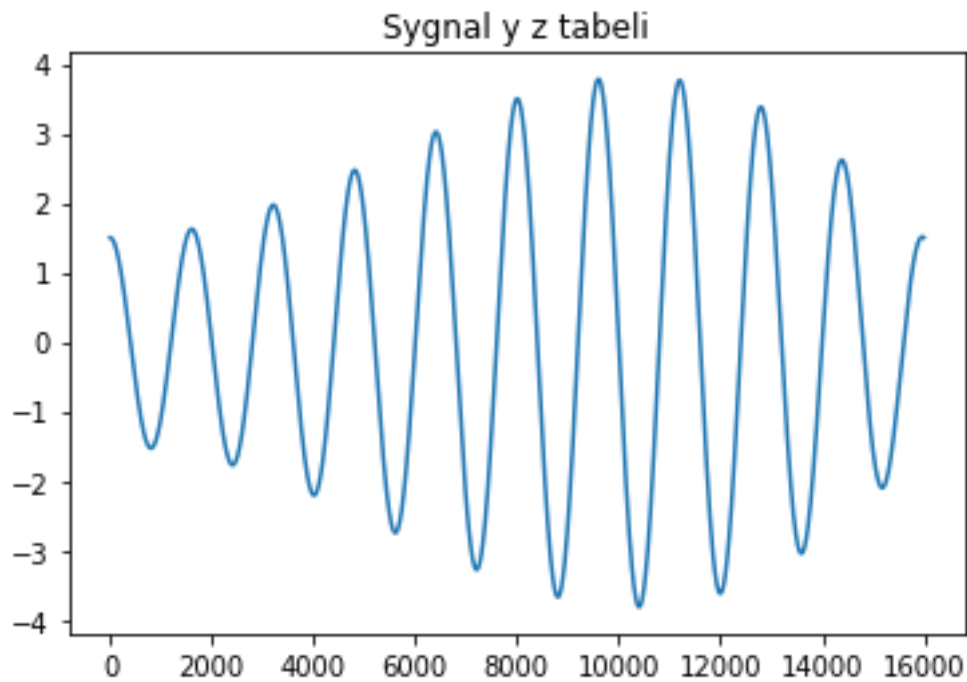
Zad 2

$$\begin{aligned}
 5. \quad & y(t) = [2 \cdot t \cdot \sin(0.5 \cdot t \cdot \pi) + 1.5] \cdot \cos(9\pi \cdot t + \pi \cdot t) \\
 & z(t) = y(t)x(t) + |x(t) + 2| \cdot [y(t)^2 + 0.32] \\
 & v(t) = \sqrt{|x(t)z(t) + 10|} \cdot (|y(t)| + 1.2) \cdot \sin(2\pi \cdot t)
 \end{aligned}$$

```

#funkcja y z tabeli
funkcja_y = lambda t: (2*t*sin(0.5*t*pi) + 1.5)*cos(9*pi*t+pi*t)
y = generate_signal(funkcja_y, f_s, T)
plt.plot(range(len(y)), y)
plt.title("Sygnal y z tabeli")

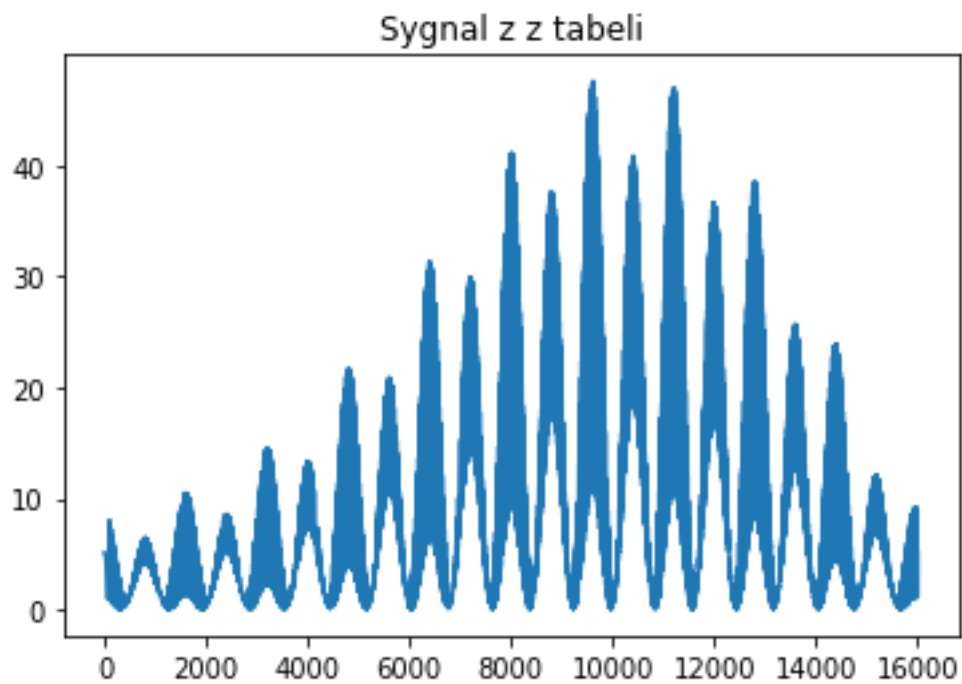
```



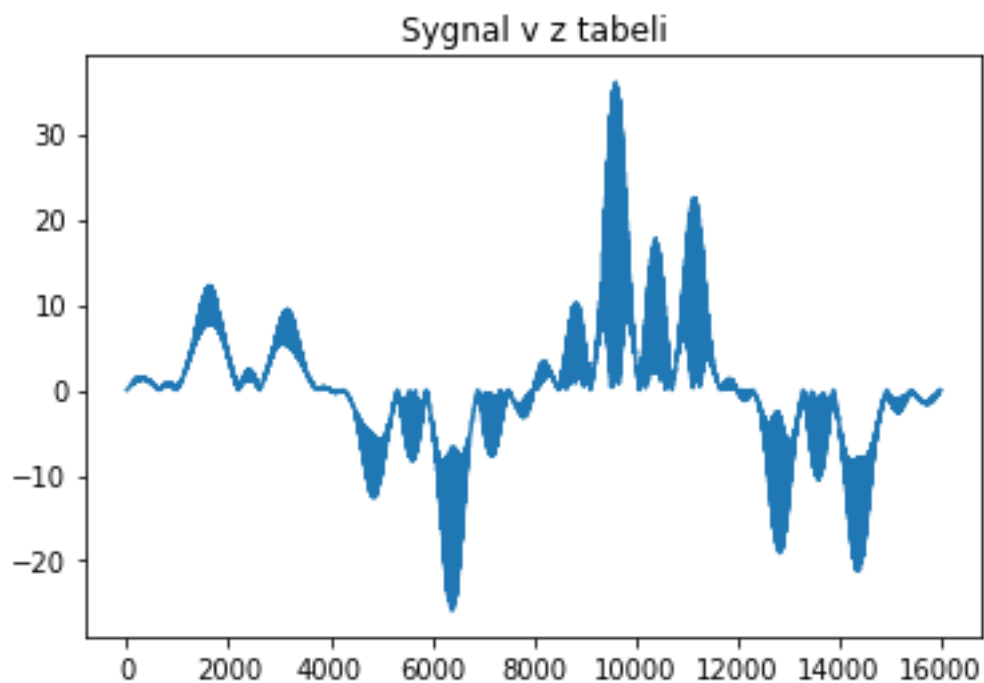
```

#funkcja z z tabeli
funkcja_z = lambda t: funkcja_x(t)*funkcja_y(t)+ abs(funkcja_x(t)+2)*(f
unkcja_y(t)**2 + 0.32)
z = generate_signal(funkcja_z, f_s, T)
plt.plot(range(len(z)), z)
plt.title("Sygnal z z tabeli")

```



```
#funkcja v z tabeli
funkcja_v = lambda t: sqrt(abs(funkcja_x(t)*funkcja_z(t)+10))*(abs(funkcja_y(t) + 1.2))*sin(2*pi*t)
v = generate_signal(funkcja_v, f_s, T)
plt.plot(range(len(v)), v)
plt.title("Sygnał v z tabeli")
```



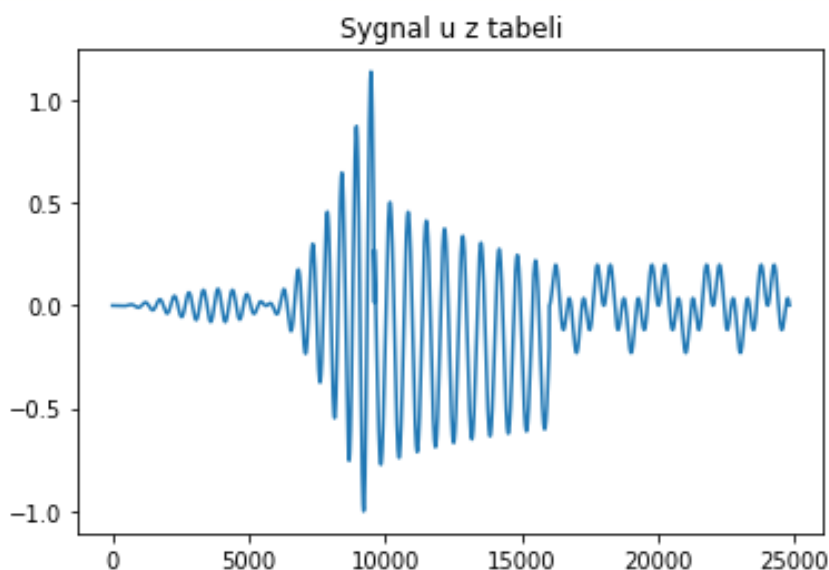
Zad 3

$$u(t) = \begin{cases} (-t^2 + 0.5) \cdot \sin(30\pi \cdot t) \cdot \log_2(t^2 + 1) & \text{dla } 1.2 > t \geq 0 \\ \frac{1}{t} \cdot 0.8 \cdot \sin(24\pi \cdot t) - 0.1t & \text{dla } 2 > t \geq 1.2 \\ |\sin(2\pi \cdot t^2)|^{0.8} & \text{dla } 2.4 > t \geq 2 \\ 0.23 \cdot \sin(20\pi \cdot t) \cdot \sin(12\pi \cdot t) & \text{dla } 3.1 > t \geq 2.4 \end{cases}$$

```
#czas trwania sygnału wynikający ze wzoru funkcji u
T_u = 3.1
```

```
def funkcja_u(t):
    '''Definicja funkcji u na przedziałach'''
    if t < 0 or t >= 3.1:
        raise Exception("Wrong time!!")
    if t < 1.2:
        return (-t**2 + 0.5)*sin(30*pi*t)*log(t**2+1,2)
    if t < 2:
        return (1/t)*0.8*sin(24*pi*t)-0.1*t
    if t < 2.4:
        return abs(sin(2*pi*t**2))**0.8
    if t < 3.1:
        return 0.23*sin(20*pi*t)*sin(12*pi*t)
```

```
#funkcja u z tabeli
u = generate_signal(funkcja_u, f_s, T_u)
plt.plot(range(len(u)), u)
plt.title("Sygnał u z tabeli")
```



Zad 4

$$b_k(t) = \sum_{h=1}^{H_k} \frac{(-1)^h}{3h^2} \cos(2\pi \cdot h \cdot t + \sin(6\pi \cdot t))$$

2, 20, 40

```
#czestotliosc dla b1, b2, b3
f_s = 22.05
#czas dla b1, b2, b3
T = 1

from functools import partial

def funkcja_b_k(t, k):
    '''definicja funkcji b_k dla k=1,2,3'''
    H = [2, 20, 40]
    return sum([(cos( 2*pi *h*t + sin(6*pi*t)))*((-1)**h)/(3*h**2) for h in range(1, H[k-1])])

#sygnaly b1 do b3
b_1 = generate_signal(partial(funkcja_b_k,k=1), f_s, T_u)
b_2 = generate_signal(partial(funkcja_b_k,k=2), f_s, T_u)
b_3 = generate_signal(partial(funkcja_b_k,k=3), f_s, T_u)

plt.plot(range(len(b_1)), b_1)
plt.plot(range(len(b_2)), b_2)
plt.plot(range(len(b_3)), b_3)
plt.title("Wspolny wykres dla sygnalow b1, b2, b3")
```

