МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”
Кафедра електронної інженерії

ЛАБОРАТОРНА РОБОТА № 1

з дисципліни «Теорія сигналів»

«Назва роботи**»**

Студента 3 курсу,
групи ДП-82

Кузьмінського О.Р.

Київ – 2020

ЛАБОРАТОРНА РОБОТА № 1

«Основи програмування мовою Python»

Мета роботи: Ознайомлення з основами програмування мовою Python на прикладі використання стандартних функцій, побудови файлів-сценаріїв та створення функцій користувача (на прикладі розв’язку системи рівнянь моделі «хижак-жертва»).

6. Ознайомитися з роботою функцій, що генерують прямокутні імпульси, гаусівські імпульси, трикутні імпульси та послідовності імпульсів заданої форми.

6.1 Побудувати одиночний прямокутний імпульс. Задати проміжок значень часу 10 секунд, частота дискретизації 256 Гц. Побудувати графік одиничного прямокутного імпульсу шириною 300 мс, з центром в момент часу 4 с.

6.2 Написати файл-сценарій для побудови графіку прямокутного імпульсу, тривалість та амплітуда якого буде задаватися з клавіатури. Розташування імпульсу задавати випадковим числом, але передбачити перевірку, чи не виходе імпульс за межі графіка;

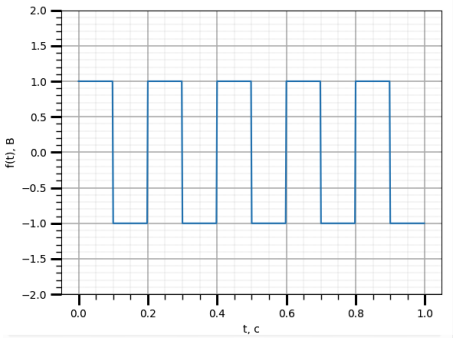
6.3 Побудувати послідовність прямокутних імпульсів для двох випадків: а) коли інтервали між імпульсами однакові, б) коли інтервали між імпульсами випадкові і задаються програмно.

7. Зберегти дані розрахунку функції в файл. Прочитати їх із файлу в іншому сценарії, побудувати графік функції.

8. Побудувати власний файл-функцію для побудови графіка синусоїдального сигналу із заданою частотою, амплітудою та тривалістю для частоти дискретизації 256 Гц. В якості вихідного параметру функції вивести середнє значення синусоїди.

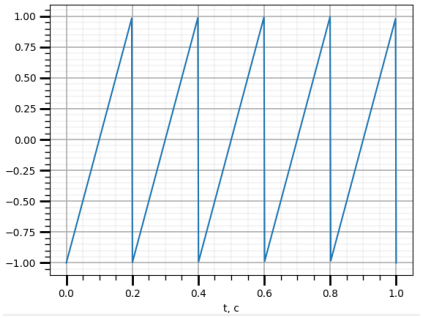
6. Прямокутний імпульс

```
t = np.linspace(0, 1, 500, endpoint=False)
plt.plot(t, signal.square(2 * np.pi * 5 * t))
plt.ylim(-2, 2)
plt.show()
```



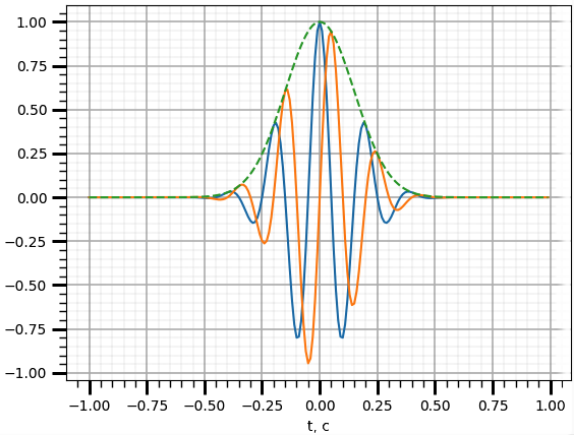
Пилкоподібний імпульс

```
t = np.linspace(0, 1, 500)
plt.plot(t, signal.sawtooth(2 * np.pi * 5 * t))
plt.show()
```



Гаусівський імпульс

```
t = np.linspace(-1, 1, 2 * 100, endpoint=False)
i, q, e = signal.gausspulse(t, fc=5, retquad=True, retenv=True)
plt.plot(t, i, t, q, t, e, '--')
plt.show()
```



6.1. Фрагмент коду.

```
t=np.linspace(0, 10, num=10000)
i=0
for el in t:
    if 3.85<=t[i]<=4.15:
        t[i]=1
    else:
        t[i]=0
    i=i+1
mp.plot(t)
mp.show()
```

Результат роботи програми

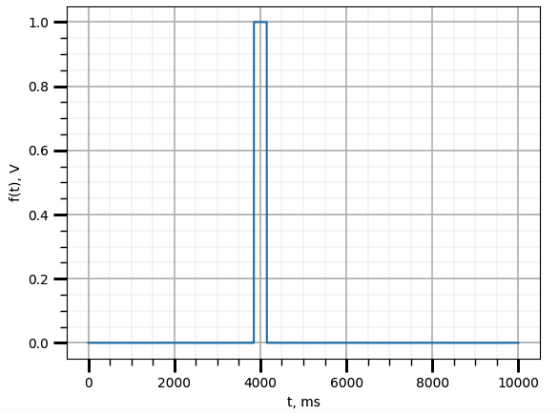


Рис.1. Одиничний прямокутний сигнал шириною 300 мс тривалістю 10 с

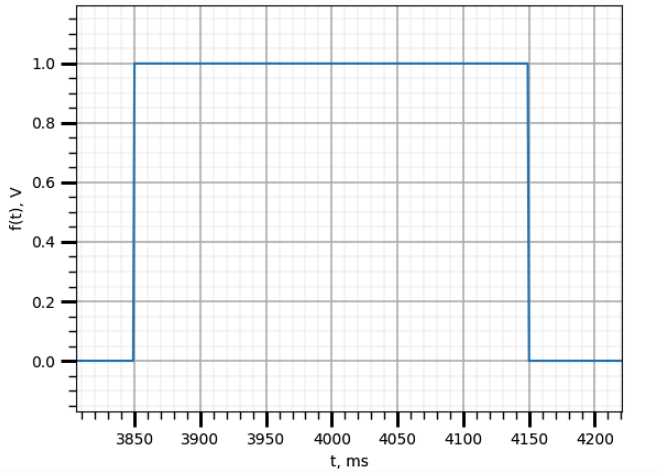
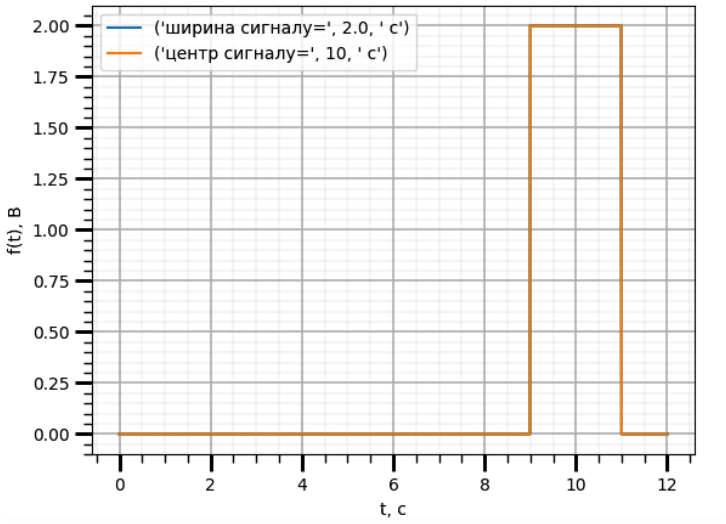


Рис.2.Одиничний прямокутний сигнал шириною 300 мс (збільшено)

6.2.Фрагмент коду:

```
Amplitude =int(input("Введіть амплітуду сигналу:"))
tn=int(input("Введіть тривалість сигналу:"))
t0 = 0
cent=randint(1,tn)
print(cent)
side=randint(cent,tn)/10
delta_t = 1.0/(tn*10) #адаптивний вибір кроку дискретизації
def function(x_depend):
    if (cent-side)<=x_depend<=(cent+side):
        return Amplitude
    return 0
x = []
y = []
time_po_osi_x = t0
while time_po_osi_x<=tn:
    x.append(time_po_osi_x)
    y.append(function(time_po_osi_x))
    time_po_osi_x += delta_t
plt.plot(x,y,'g')
ax.plot(x,y,label=('ширина сигналу=' ,(cent+side)-(cent-side),' c'))
ax.plot(x,y,label=('центр сигналу=' ,cent,' c'))
ax.legend()
plt.show()
```

Результат роботи програми:



6.3.Фрагмент коду:

```
T=10

N=10

x = np.linspace(0, T*N, 10000, endpoint=False)

y=signal.square(2 * np.pi * (1/T) * x + (2*np.pi)/4)

plt.plot(x,y)

plt.ylim(-2, 2)

plt.xlim(0, T*N)

plt.show()
```

Результат роботи програми: