# Практическая работа 3. Анализ данных на языке Python

**ФИО: Пестов Андрей Витальевич**

**Группа: R4136c (КСИТ\_1.1)**

## ЗАДАНИЕ

Есть набор данных, который содержит данные о напряжении, токе и времени измерения. Период измерения 0,001 с. Период тестового сигнала 0,1 с. Количество периодов тестового сигнала - 1000. Тип двигателя - двигатель постоянного тока.

**Этапы работы:**

1. Импортировать библиотеки в Python.

2. Загрузить и подготовить данные.

3. Нарисовать графики тока и напряжения.

4. Рассчитать значения параметров L и R.

5. Рассчитать средние значения и стандартное отклонение значений параметров L и R.

**1. Импортировать библиотеки.**

import numpy as npy

from numpy import linalg as lnlg

import matplotlib.pyplot as mpl

**2. Загрузка и подготовка данных**

data\_read = npy.genfromtxt('testLab1Var40.csv', delimiter=',')

time = data\_read[:, 0]

time = time[:, npy.newaxis]

current = data\_read[:, 1]

current = current[:, npy.newaxis]

voltage = data\_read[:, 2]

voltage = voltage[:, npy.newaxis]

Запишите значение 11-го элемента массива напряжений и временного массива.

**Ответ:** voltage[10] = **1** time[10]=**0.011**

**3. Нарисовать графики тока и напряжения**

fig, (ay1, ay2) = mpl.subplots(2, 1, sharex=True)

T\_per = 0.1

ay1.plot(time[time < 2\*T\_per], voltage[time < 2\*T\_per])

ay1.grid()

ay1.set\_xlabel('time, s')

ay1.set\_ylabel('voltage, V')

ay2.plot(time[time < 2\*T\_per], current[time < 2\*T\_per])

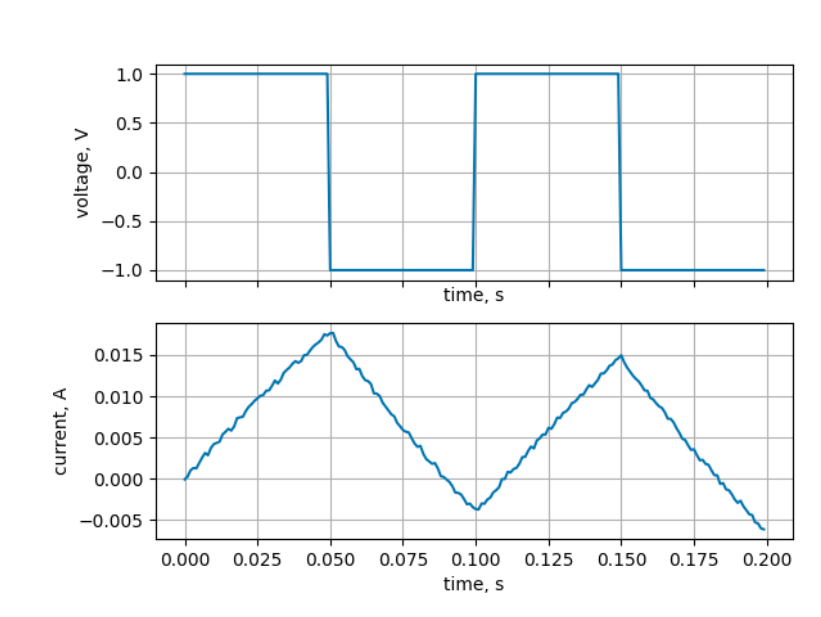
ay2.grid()

ay2.set\_xlabel('time, s')

ay2.set\_ylabel('current, A')

mpl.show()

fig.savefig('Recieved data(part)')



**Рисунок 1.** Два периода сигналов напряжения и тока соответственно.

**4. Расчёт значений параметров L и R.**

X = npy.concatenate([voltage[0:len(voltage) -

2], current[0:len(current)-2]], axis=1)

Y = current[1:len(current)-1]

K = npy.dot(npy.dot(lnlg.inv(npy.dot(X.T, X)), X.T), Y)

Td = 0.001

R = 1 / K[0] \* (1 - K[1])

T = -Td / npy.log(K[1])

L = T\*R

current\_est = X.dot(K)

fig, ax = mpl.subplots(1, 1)

mpl.plot(time[time < T\_per], current[time < T\_per])

mpl.plot(time[time < T\_per], current\_est[time[0:len(current)-2] < T\_per])

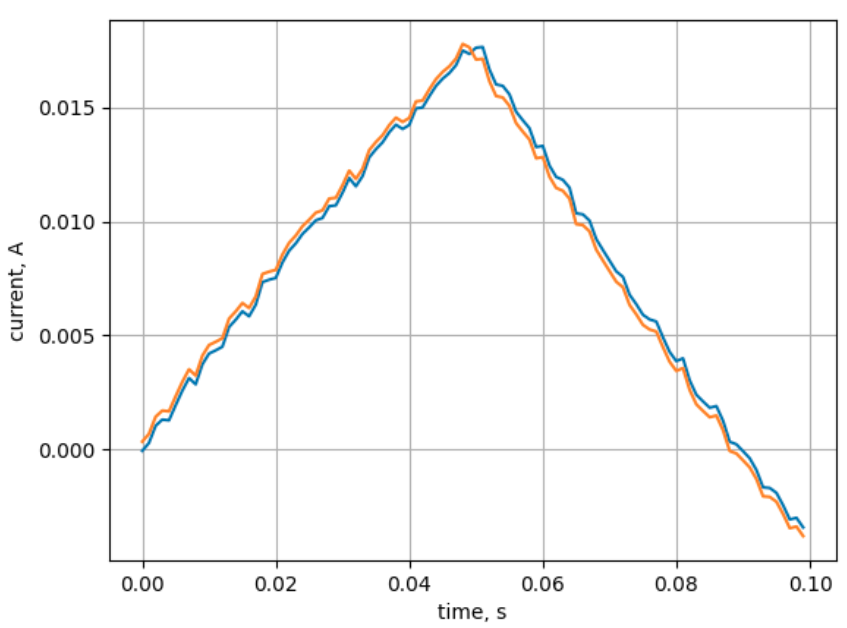
ax.grid()

ax.set\_xlabel('time, s')

ax.set\_ylabel('current, A')

mpl.show()

fig.savefig('Compared data(part)')



**Рисунок 2.** Реальные и расчетные графики тока.

**Ответ:** расчетные значения сопротивления и индуктивности: R =16.45595174, L =2.4683957.

**5. Расчёт средних значений и стандартного отклонения значений параметров L и R.**

n = 1000

for i in range(0, n-1, 1):

ind = (time >= T\_per\*i) & (time <= T\_per\*(i+1))

new\_current = current[ind]

new\_current = new\_current[:, npy.newaxis]

new\_voltage = voltage[ind]

new\_voltage = new\_voltage[:, npy.newaxis]

X = npy.concatenate([new\_voltage[1:len(new\_voltage) - 1],

new\_current[0:len(new\_current) - 2]], axis=1)

Y = current[1:len(new\_current) - 1]

K = npy.dot(npy.dot(lnlg.inv(npy.dot(X.T, X)), X.T), Y)

if K[1] > 0:

R = 1/K[0]\*(1-K[1])

T = -Td / npy.log(K[1])

R\_est.append(R)

L\_est.append(T\*R)

R\_est = npy.array(R\_est)

L\_est = npy.array(L\_est)

print('Mean value of R: ', npy.mean(R\_est), ' Ohm')

print('Standart deviation of R: ', npy.std(R\_est))

print('Mean value of L = ', npy.mean(L\_est), ' Hn')

print('Standart deviation of R: ', npy.std(L\_est))

**Таблица 1.**

|  |  |  |
| --- | --- | --- |
|  | **R** | **L** |
| **Среднее значение** | -22.297 Ohm | 0.906 Hn |
| **Стандартное отклонение** | 31.713  31.713 | 0.23 |