**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информатика»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3

по дисциплине «Математические модели сложных систем»

## на тему: «Построение моделей по экспериментальным данным»

Выполнил: студент гр. ИП-31

Коваленко А.И.

Приняла: доцент

Трохова Т.А.

Гомель 2023

**Цель работы:** получить навыки разработки моделей по результатам эксперимента, применить функции аппроксимации, выполнить исследования по моделям.

***Часть 1. Формирование траектории движения мобильного робота***

## **Постановка задачи моделирования**

1. Ввести координаты опорных точек для фиксации обязательного захода в них робота.

2. Подобрать аппроксимирующую аналитическую функцию для вычисления зависимости Y(X) движения мобильного робота в плоскости.

3. Построить график движения мобильного робота по рассчитанной траектории.

4. Найти максимальное отклонение по координате Y полученной траектории движения от опорной точки.

5. Найти расстояние, пройденное роботом от исходной точки до конечной, если он движется по траектории Y(X).

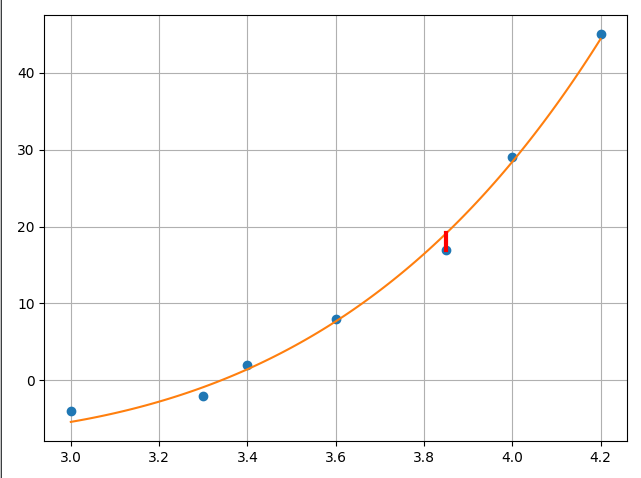
|  |  |  |  |
| --- | --- | --- | --- |
|  |  | 3;3.3;3.4;3.6;3.85;4;4.2 | -4;-2; 2; 8; 17; 29; 45 |

**Листинг:**

**Задание 1:**

import numpy as np  
import matplotlib.pyplot as plt  
from scipy.optimize import curve\_fit  
  
  
def func(x, a, b):  
 return a \* np.exp(x) + b \* x \*\* 2  
  
  
def max\_error\_search():  
 max\_error = np.abs(func(X[0], a, b) - Y[0])  
 index = 0  
 for i in range(1, len(X)):  
 if np.abs(func(X[i], a, b) - Y[i]) > max\_error:  
 max\_error = np.abs(func(X[i], a, b) - Y[i])  
 index = i  
 print('Максимальное отклонение по координате Y полученной траектории '  
 f'движения от опорной точки: {max\_error:.3f} и значение Х {X[index]:.3f}')  
 plt.plot([X[index], X[index]], [Y[index], func(X[index], a, b)], color='red', linewidth='3')  
  
  
def distance\_going\_robot():  
 distance\_list = [np.sqrt((x\_k[i] - x\_k[i - 1]) \*\* 2  
 + (y\_k[i] - y\_k[i - 1]) \*\* 2) for i in range(1, len(y\_k))]  
 distance = sum(distance\_list)  
 print('Расстояние, пройденное роботом от исходной точки до конечной,'  
 f' если он движется по траектории Y(X): {distance}')  
  
  
X = np.array([3, 3.3, 3.4, 3.6, 3.85, 4, 4.2])  
Y = np.array([-4, -2, 2, 8, 17, 29, 45])  
plt.plot(X, Y, "o")  
  
K, S = curve\_fit(func, X, Y)  
  
a = K[0]  
b = K[1]  
  
x\_k = np.arange(3.0, 4.21, 0.01)  
y\_k = func(x\_k, a, b)  
plt.plot(x\_k, y\_k)  
  
max\_error\_search()  
distance\_going\_robot()  
  
plt.grid()  
plt.show()

**Результат выполнения задания 1**





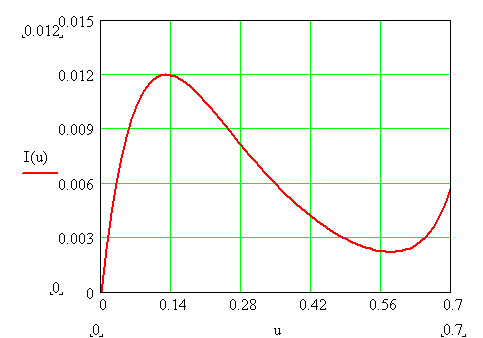
***Часть 2. Компьютерная модель туннельного диода (синтез робастной модели)***

## **Постановка задачи моделирования**

1. Ввести координаты опорных точек вольт-амперной характеристики, используя график приложения А.

2. Подобрать аппроксимирующую аналитическую функцию I(u) для заданной графически вольт-амперной характеристики.

3. Построить график исходной и аппроксимирующей характеристик.

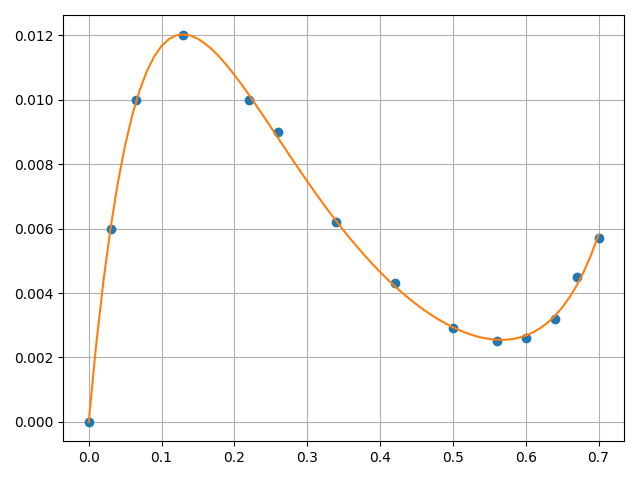


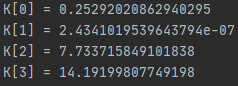
**Листинг:**

**Задание 2:**

def func(u, a, d, alpha, betta):  
 return a \* u \* np.e \*\* (-alpha \* u) + d \* (np.e \*\* (betta \* u) - 1)  
  
  
alpha\_n = 7  
betta\_n = 10  
a\_n = 0.2  
d\_n = 1e-7  
  
X = np.array([0, 0.03, 0.065, 0.13, 0.22, 0.26, 0.34, 0.42, 0.5, 0.56, 0.6, 0.64, 0.67, 0.7])  
Y = np.array([0, 0.006, 0.01, 0.012, 0.01, 0.009, 0.0062,  
 0.0043, 0.0029, 0.0025, 0.0026, 0.0032, 0.0045, 0.0057])  
  
K, S = curve\_fit(func, X, Y, (a\_n, d\_n, alpha\_n, betta\_n))  
  
a = K[0]  
d = K[1]  
alpha = K[2]  
betta = K[3]  
  
x\_k = np.arange(0, 0.71, 0.01)  
y\_k = func(x\_k, a, d, alpha, betta)  
  
for i in range (len(K)):  
 print(f'K[{i}] = {K[i]}')  
  
plt.plot(X, Y, "o")  
plt.plot(x\_k, y\_k)  
plt.grid()  
plt.show()

**Результат выполнения задания 2**





**Вывод:** получил навыки по построению моделей по экспериментальным данным.