**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 **«Математическое моделирование сложных систем»**

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

Выполнил: студент гр. ИП-31

Кузнецова Е. А.

Принял: доцент

Трохова Т. А.

Дата сдачи отчета: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата допуска к защите: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Гомель 2024

**Цель работы:** получить навыки разработки моделей по результатам

эксперимента, применить функции аппроксимации, выполнить исследования по моделям.

**Задание 1.** ***Моделирование траектории движения мобильных роботов***

**Постановка задачи моделирования**

Два робота движутся по разным траекториям, фиксируя координаты опорных точек пройденного пути.

Необходимо:

1. Ввести координаты опорных точек движения роботов.

2. Подобрать аппроксимирующие аналитические функции траекторий

движения мобильных роботов по индивидуальным вариантам (таблица 2).

3. Построить график движения мобильных роботов по рассчитанным

траекториям. На графике маркерами должны быть помечены координаты

исходных точек и сплошной линией – аппроксимирующих функций.

4. Выполнить исследования по построенной модели:

- рассчитать прогнозные траектории движения роботов до заданного

значения xf по оси Х;

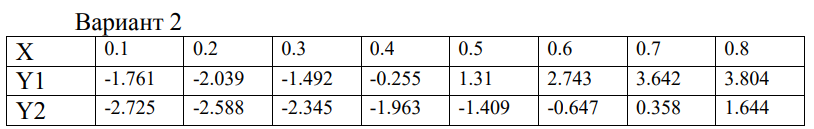
- спрогнозировать координаты точки пересечения траекторий движения

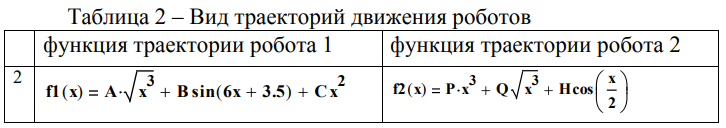
роботов;

- найти расстояние, пройденное каждым роботом до финальной точки (xf, yf).

5. Нанести результаты исследований на отдельный график.

Таблица 1 - Координаты опорных точек движения роботов

****

****

**Код программы:**

import numpy as np

import matplotlib.pyplot as plt

from scipy.optimize import curve\_fit

from scipy.integrate import quad

x = np.array([0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8])

y1 = np.array([-1.761, -2.039, -1.492, -0.255, 1.31, 2.743, 3.642, 3.804])

y2 = np.array([-2.725, -2.588, -2.345, -1.963, -1.409, -0.647, 0.358, 1.644])

XF = 1.1

def func1(x, A, B, C):

return A \* np.sqrt(x\*\*3) + B \* np.sin(6\*x + 3.5) + C\*(x\*\*2)

def func2(x, P, Q, H):

return P\*x\*\*3 + Q\*np.sqrt(x\*\*3) + H\*np.cos(x/2)

#от 0.1 до 0.8, потом расширяем до 1.1

popt1, \_ = curve\_fit(func1, x, y1)

popt2, \_ = curve\_fit(func2, x, y2)

print("Oптимальные параметры для функции:")

print(popt1)

print(popt2)

#создать новый массив

X\_fit = np.linspace(0.1, XF, 100)

y1\_new = func1(X\_fit, \*popt1)

y2\_new = func2(X\_fit, \*popt2)

#точка

idx = np.argwhere(np.diff(np.sign(y1\_new - y2\_new))).flatten()

x\_cross, y\_cross = X\_fit[idx], y1\_new[idx]

def integrand1(x):

return np.sqrt(1 + (func1(x, \*popt1))\*\*2)

def integrand2(x):

return np.sqrt(1 + (func2(x, \*popt2))\*\*2)

#длина кривых

length1, \_ = quad(integrand1, 0.1, XF)

length2, \_ = quad(integrand2, 0.1, XF)

print("Длина кривой для функции 1:", length1)

print("Длина кривой для функции 2:", length2)

plt.plot(x, y1, 'ro', label='Robot 1 Points')

plt.plot(x, y2, 'bo', label='Robot 2 Points')

plt.plot(X\_fit, func1(X\_fit, \*popt1), 'r-', label='Robot 1 Fit')

plt.plot(X\_fit, func2(X\_fit, \*popt2), 'b-', label='Robot 2 Fit')

plt.plot(x\_cross, y\_cross, 'go')

plt.xlabel('X')

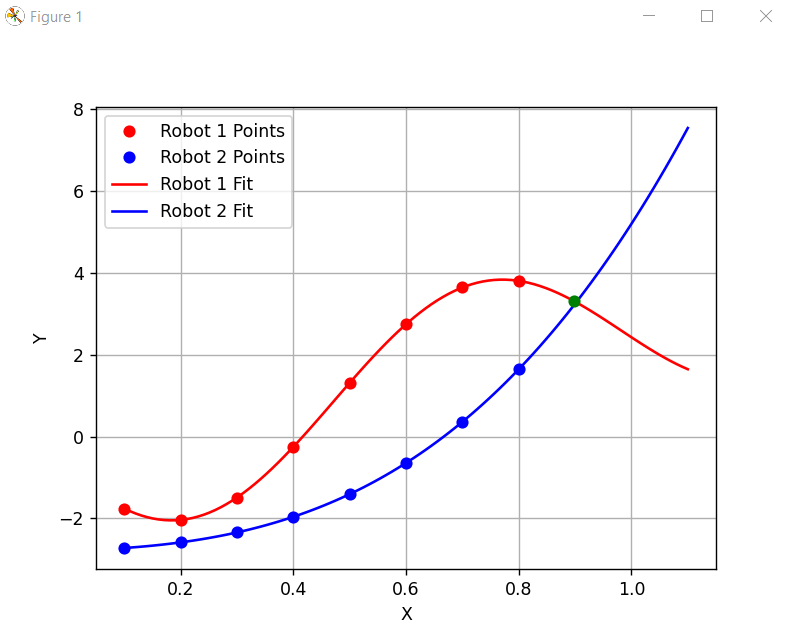
plt.ylabel('Y')

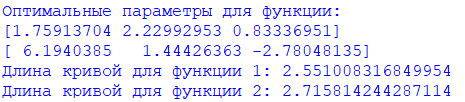
plt.legend()

plt.grid(True)

plt.show()

**Результат выполнения:**





**Вывод:** в результате выполнения лабораторной работы мноюполучены навыки разработки моделей по результатам эксперимента, применены функции аппроксимации, выполнено исследования по моделям.