**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 № 1

по дисциплине «Алгоритмы и структуры данных»

на тему: «Линейные динамические структуры»

Выполнил: студент гр. ИП-31

Казутин П. Н.

Принял: ст. преподаватель

Косинов Г. П.

Гомель 2021

**Цель работы:** изучить основы работы с линейными динамическими структурами (списками, очередями, стеками, кольцевыми буферами и т.д.).

**Практическая часть:**

## Задача 1

## Постановка задачи моделирования

1. Считать с диска заданный файл исходных данных, сформировать вектор временного интервала, построить график функции ЭДС в зависимости от времени.
2. Выделить участок исходной функции от максимального до минимального значений, аппроксимировать его с помощью полиномиальной регрессии. Построить график исходной и аппроксимирующей функции

### Исходные данные для моделирования

Задан файл результатов эксперимента в числовом виде, задано время проведения эксперимента.

|  |  |  |
| --- | --- | --- |
| N варианта | Имя файла | ***Время исследований*** в мс |
| 2 | tan2.dat | 8 |

Листинг кода:

**def** task1():  
 print(**"Лабораторная работа №3"**)  
 source = np.loadtxt(**'tan2.txt'**)  
 dt = np.linspace(0, 8, source.\_\_len\_\_())plt.grid()n\_max, n\_min = np.argmax(source), np.argmin(source)  
 print(**f'(max\_n, min\_n): {**n\_max, n\_min**}'**)  
 data\_between = source[n\_max: n\_min]  
 dt\_b = dt[n\_max: n\_min]print(data\_between.\_\_len\_\_())  
 reduced\_data = []  
 reduced\_time = []  
 **for** i **in** range(0, data\_between.\_\_len\_\_()):  
 **if** i % 29 == 0:  
 reduced\_time.append(dt\_b[i])  
 reduced\_data.append(data\_between[i])  
 print(reduced\_time)  
 plt.scatter(reduced\_time, reduced\_data)k = np.polyfit(reduced\_time, reduced\_data, 5)  
 print(**f'Коэффиценты: {**k**}'**)  
 f\_reduced = np.poly1d(k)  
 new\_time = np.arange(dt\_b[0], dt\_b[-1], 0.01)  
 new\_values = f\_reduced(new\_time)  
 plt.plot(new\_time, new\_values)  
 plt.show()

Результат выполнения:

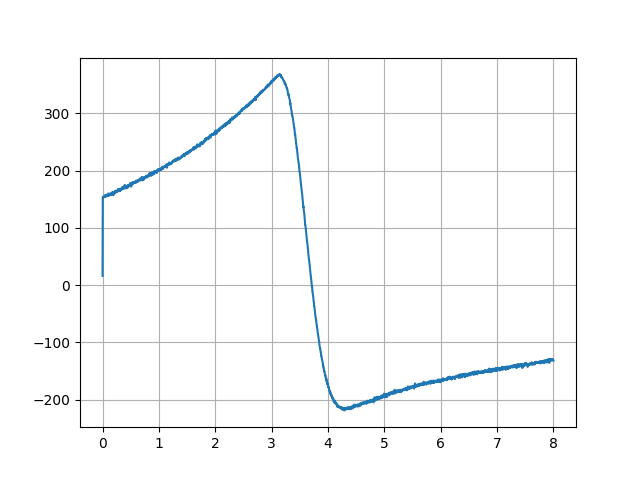
Лабораторная работа №3

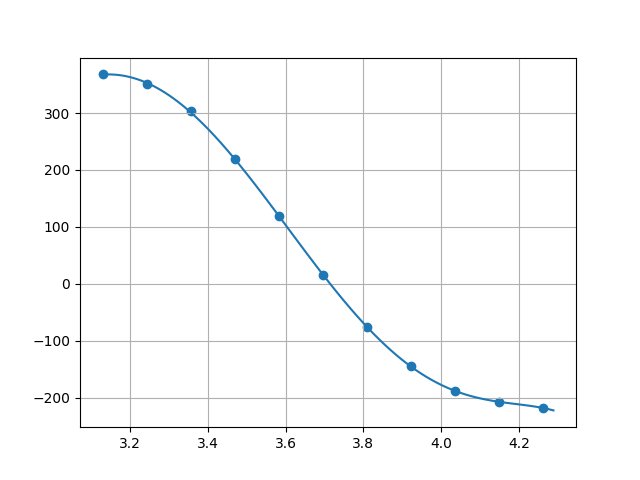
(max\_n, min\_n): (801, 1099)

[3.12890625, 3.2421875, 3.35546875, 3.46875, 3.58203125, 3.6953125, 3.80859375, 3.921875, 4.03515625, 4.1484375, 4.26171875]

Коэффиценты: [ -1228.46074946 21889.1261309 -154273.4444897 537176.39542347

-923803.67289766 628042.68597077]





**Вывод**: в ходе выполнения первого задания построил график изменения ЭДС, вырезал участок между экстремумами, аппроксимировал этот участок полиномом 5-й степени с коэф-ми: [-1228.46, 21889.13, -154273.44, 537176.39, -923803.67, 628042.68].

***Часть 2. 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.

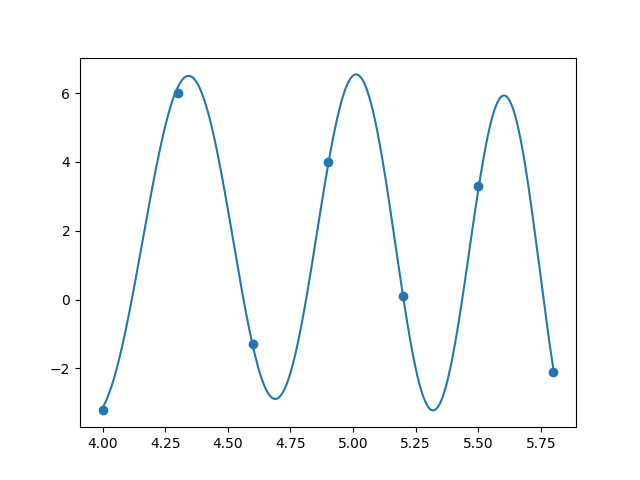
4. Найти максимальное отклонение по координате Y полученной траектории движения от опорной точки.

|  |  |  |  |
| --- | --- | --- | --- |
| 3 |  | 4;4.3;4.6 ;4.9; 5.2; 5.5; 5.8 | -3.2;6;-1.3;4;0.1;3.3;-2.1 |

Листинг кода:

**def** task2():  
 print(**'Задание №2'**)  
 func = **lambda** x, a, b: a \* np.cos(x \*\* 2) + b \* np.sin(x)  
 x = [4, 4.3, 4.6, 4.9, 5.2, 5.5, 5.8]  
 y = [-3.2, 6, -1.3, 4, 0.1, 3.3, -2.1]  
 plt.scatter(x, y)  
 *# получаем через curve\_fit примерную фунуцию* coef, err = opt.curve\_fit(func, x, y, (0.0, 0.0))  
 print(**f'Коэфф-ы: {**coef**}'**)  
 dx = np.arange(x[0], x[-1], 0.001)  
 dy = func(dx, coef[0], coef[1])  
 plt.plot(dx, dy)  
 plt.show()  
 *# отклонение* ddy = func(np.array(x), coef[0], coef[1])  
 print(**f'Отклонение: {**np.abs(ddy - y)**}'**)  
 print(**f'Максимальное отклонение: {**np.max(np.abs(ddy - y))**}'**)

Результат выполнения:



Задание №2

Коэфф-ы: [ 4.75604589 -1.87125887]

Отклонение: [0.06150094 0.16629546 0.04625379 0.10119388 0.01703603 0.1063493 0.07870417]

Максимальное отклонение: 0.16629546013959562

**Вывод**: в ходе выполнения задания разработал программу, по выданным точкам подобрал коэффициенты аппроксимирующей функции [ 4.76,-1.87], нашёл отклонения по оси игрек выданных точек от полученной функции, максимальное отклонение составило 0.166.

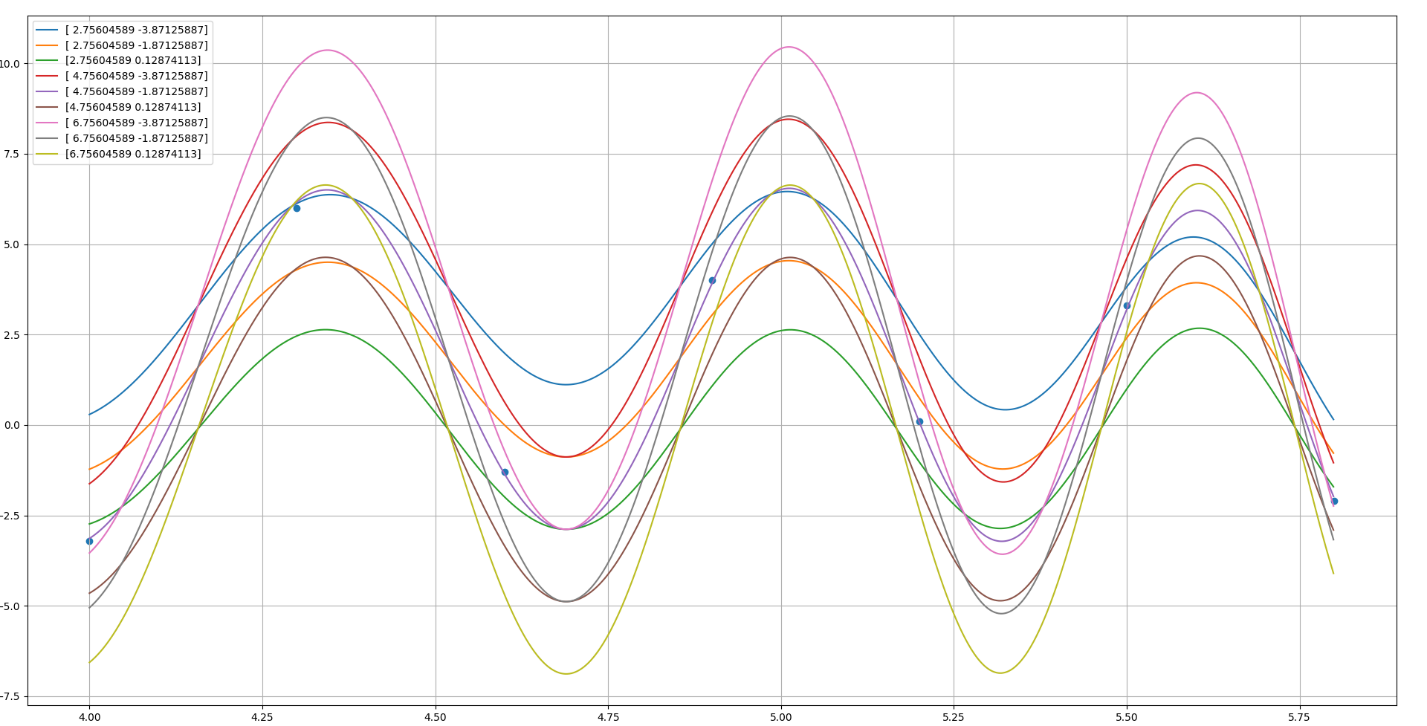
**Часть 3. Обработка результатов эксперимента по компьютерной модели**

## Постановка задачи моделирования

1. На основании статической модели, разработанной в лабораторной работе №2 для технического объекта по индивидуальному варианту, провести исследования с изменением варьируемого параметра, обобщить результаты исследований, сделать графическую интерпретацию результатов
2. Подобрать аппроксимирующую зависимость к экспериментальным данным, сделать выводы по работе.

Листинг кода:

**def** task3():  
 print(**'Задание №3'**)  
 func = **lambda** x, a, b: a \* np.cos(x \*\* 2) + b \* np.sin(x)  
 x = [4, 4.3, 4.6, 4.9, 5.2, 5.5, 5.8]  
 y = [-3.2, 6, -1.3, 4, 0.1, 3.3, -2.1]  
 plt.scatter(x, y)  
 coef, err = opt.curve\_fit(func, x, y, (0.0, 0.0))  
 dx = np.arange(x[0], x[-1], 0.001)  
 da = np.linspace(coef[0] - 2, coef[0] + 2, 3)  
 db = np.linspace(coef[1] - 2, coef[1] + 2, 3)  
 dot = np.array(list(itertools.product(da, db)))  
 xs = []  
 **for** i **in** dot:  
 xs.append([dx, func(dx, i[0], i[1])])  
 **for** i **in** xs:  
 plt.plot(i[0], i[1])  
 plt.legend(dot)  
 plt.grid()  
 plt.show()



Вывод: варьируя параметры a, b, получил 9 вариантов графика (3 варианта а \* 3 варианта b), на графике отобразил функции с варьируемыми параметрами.