

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**

Институт искусственного интеллекта

Кафедра проблем управления

**ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ**

**По курсу «Управление интеллектуальными роботами и робототехническими системами»**

|  |  |  |
| --- | --- | --- |
| Студент группы  КРБО-03-19 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Чепижный Е.А. |
|  |  |
|  |  |
|  |  |  |
| Преподаватель: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Голубов В.В.  «\_\_» \_\_\_\_\_\_ 2023 г. |
|  |  |  |

Москва 2023

Для начала работы необходимо импортировать две библиотеки:

* numpy
* matplotlib
* opencv

Далее необходимо открыть файл и прочитать данные из файла, назначенные на переменную lines, чтобы использовать ее в функции сортировки данных.



Следующим действием назначаются два массива в каждый из которых будут записываться данные, которые в основном файле перемешаны и представляют собой структуру представленную ниже на Рис.1



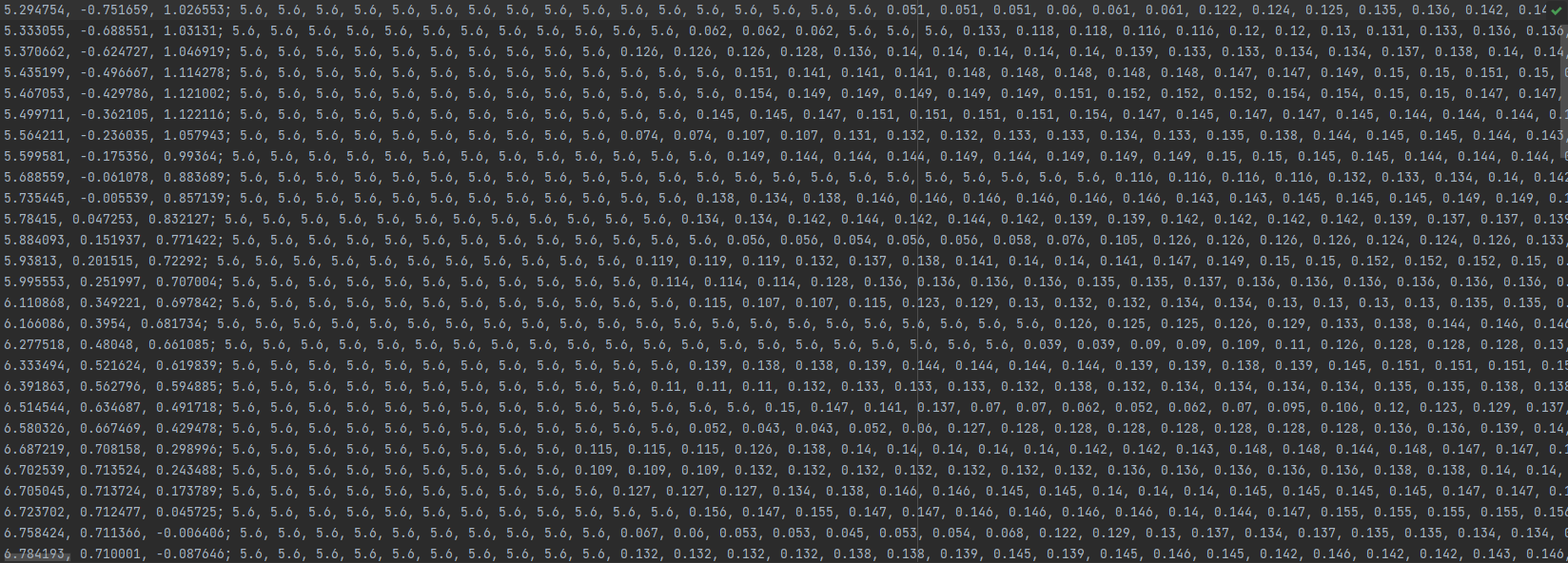
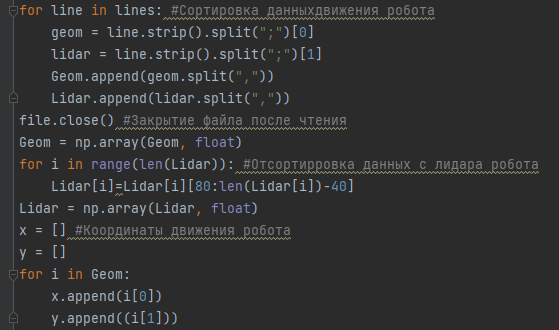


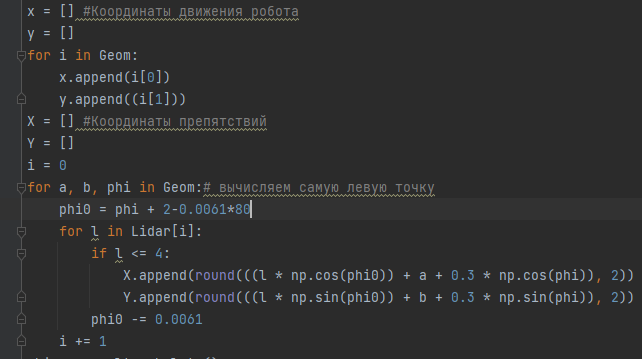
Рис. 1 - данный исходного файла

Т.к. в файле с данными находятся как одометрия, так и данные с лидара, при помощи функции for и использования split данные разделяются и записываются в два массива данных.

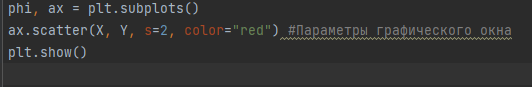
Сразу после этого при помощи той же функци for и функции len, которая подсчитывает количество символов в строке, перезаписываем данные в массиве для отрисовки препятствий, которые заметил лидар.

Далее создаются массивы x и y, необходимые для записи данный одометрии и постройки примерного пути робота.

При помощи функции for, в которой задаются параметры a, b (аналоги x и y) и угол phi, берущиеся из одометрического массива, находим примерный первый угол, с которого начинает считаться препятствия и при помощи данных из массива лидара осуществляем отрисовку всех препятствий на пути следования робота.



Далее при помощи угла phi и переменной ax, subploat(позволяет создать много графиков) и scatter(создает графики разброса), создаем множество накладываемых друг на друга графиков в итоге на Рис.2 представлена карта препятствий, собранная при помощи данных с лидара робота



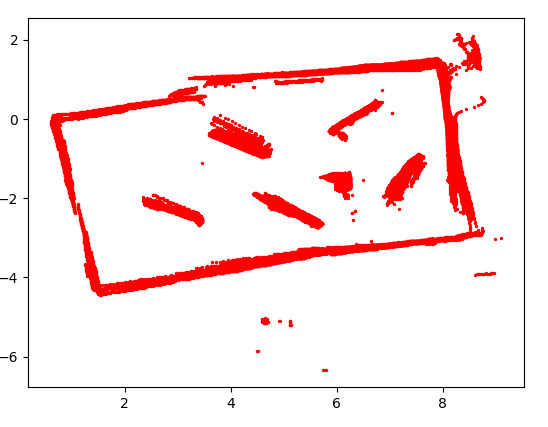


Рис. 2 - Результат работы программы

**Листинг программы:**

import matplotlib.pyplot as plt

import numpy as np

file = open("examp2.txt") #Открытие файла

lines = file.readlines() #Чтение данных из файла

Geom = [] #массив для даннных о положении робота

Lidar = [] #массив для данных о положении препятсвий

for line in lines: #Сортировка данныхдвижения робота

geom = line.strip().split(";")[0]

lidar = line.strip().split(";")[1]

Geom.append(geom.split(","))

Lidar.append(lidar.split(","))

file.close() #Закрытие файла после чтения

Geom = np.array(Geom, float)

for i in range(len(Lidar)): #Отсортирровка данных с лидара робота

Lidar[i]=Lidar[i][80:len(Lidar[i])-40]

Lidar = np.array(Lidar, float)

x = [] #Координаты движения робота

y = []

for i in Geom:

x.append(i[0])

y.append((i[1]))

X = [] #Координаты препятствий

Y = []

i = 0

for a, b, phi in Geom:# вычисляем самую левую точку

phi0 = phi + 2-0.0061\*80

for l in Lidar[i]:

if l <= 4:

X.append(round(((l \* np.cos(phi0)) + a + 0.3 \* np.cos(phi)), 2))

Y.append(round(((l \* np.sin(phi0)) + b + 0.3 \* np.sin(phi)), 2))

phi0 -= 0.0061

i += 1

phi, ax = plt.subplots()

ax.scatter(X, Y, s=2, color="red") #Параметры графического окна

plt.show()