

1)Импортируем нужные библиотеки

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import math
.
```

Рис.1.Библиотеки

2)Разделяем данные на до точки запятой, и на после точки запятой. Первые три координаты – координаты лидара, остальные данные – расстояние до стены.

```
data = pd.read_csv('/content/drive/MyDrive/examp10.txt', sep=";", header=None)
#print(data)
coord = data[0].str.split(",", expand=True)
lidar = data[1].str.split(",", expand=True)
```

Рис.2.Работа с данными

3)Координаты x,y и угол поворота преобразуем из строкового формата в числовой, чтобы можно было в дальнейшем использовать в вычислениях.

```
14 X = pd.to_numeric(coord[0])
15 Y = pd.to_numeric(coord[1])
16 W = pd.to_numeric(coord[2])
```

Рис.3.Третий пункт

4)

```
22 ang=0.5
23 for i in range(0,len(coord[0])):
24     xdir.append(X[i] * math.cos(ang) - Y[i] * math.sin(ang))
25     ydir.append(X[i] * math.sin(ang) + Y[i] * math.cos(ang))
26 |
```

Рис.4.Код поворота траектории

Данные строчки производят поворот траектории на угол альфа(ang=0.5).

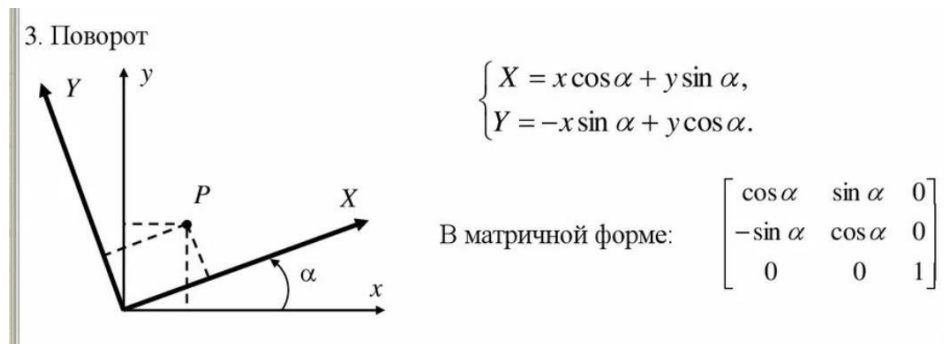


Рис.5.Формула поворота

Далее по коду происходит поворот и карты тоже на тот же угол ang.

```
50 for i in range(len(xlist)):
51     x1 = xlist[i]
52     y1 = ylist[i]
53     xlist[i] = x1 * math.cos(ang) + y1 * math.sin(ang)
54     ylist[i] = - x1 * math.sin(ang) + y1 * math.cos(ang)
```

Рис.6.Поворот карты по всё той же формуле

Это всё нужно для того, чтобы картинка легко воспринималась, потому, что без этих преобразований она выглядит так(криво под углом):

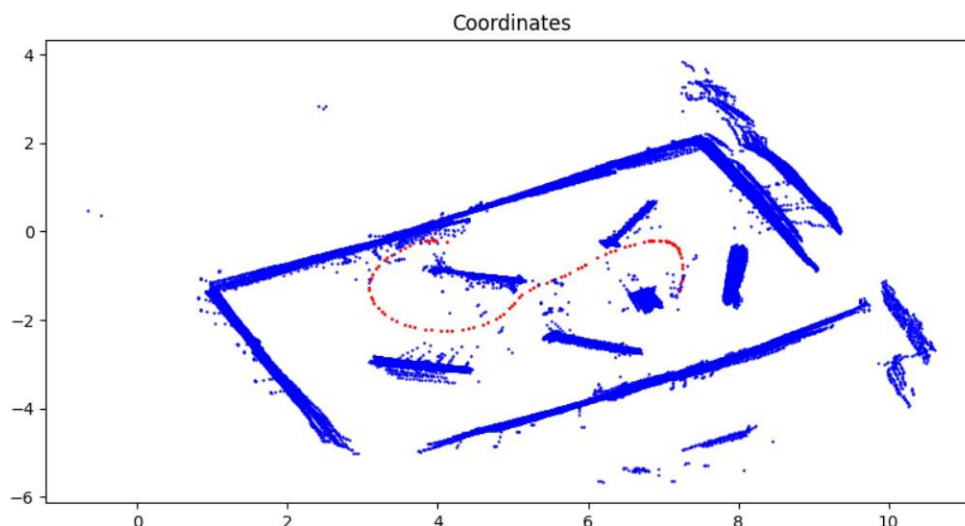


Рис.7.Карта без поворота

После поворота она стала лучше:

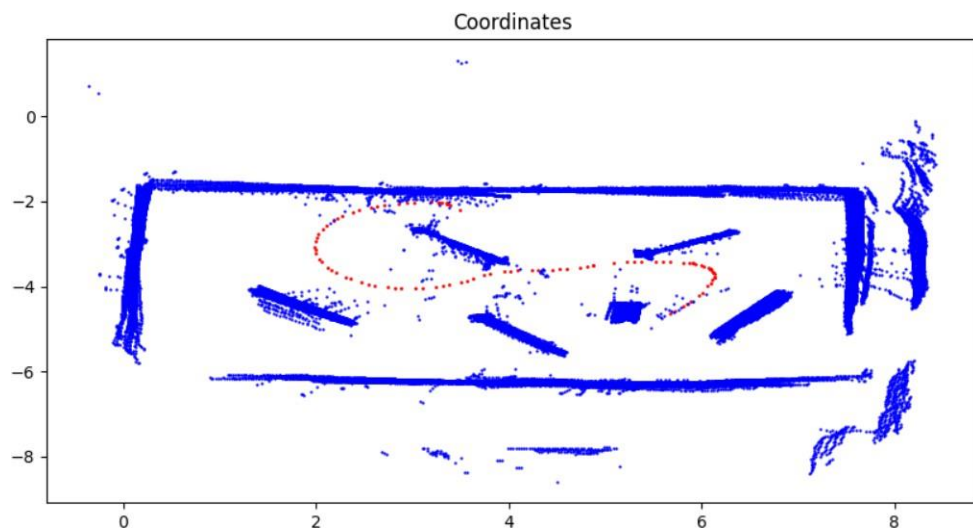


Рис.8.Карта с поворотом

5)

```

34 for j in range(0, len(lidar[0])) :
35     phi = 2.0944 + W[j]
36     for i in range(680):
37         r = float(lidar[i][j])
38         if ( r < 5.6 and r > 0.3):
39             xw = (r) * math.cos(phi) + 0.3
40             yw = r * math.sin(phi)
41             x0 = (X[j] + xw + 0.3 * math.cos(W[j]))
42             y0 = (Y[j] + yw + 0.3 * math.sin(W[j]))
43             xlist.append(x0)
44             ylist.append(y0)
45             phi = phi - 0.006152716

```

Рис.9.Создание координат для отрисовки карты

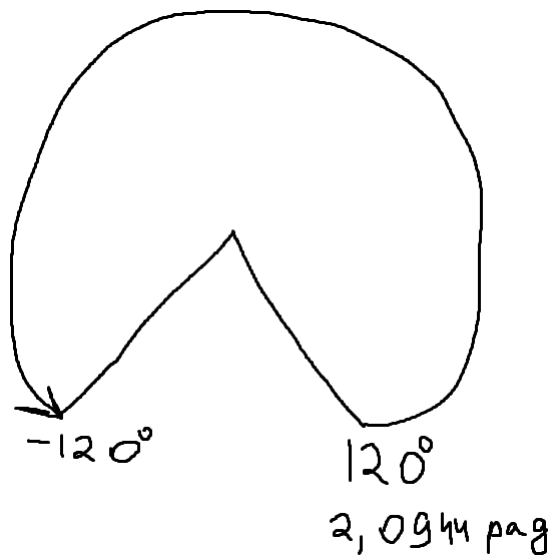


Рис.10. Угол обзора лидара

В 34-й строке проходимся по количеству координат (от нуля до 99).

В 35-й берем $\phi = 120$ градусов(в радианах) + поворот робота(в радианах), рис.10.

В 36-й строчке проходимся по столбцам от 0 до 680(в сумме 681 как и в example.10).

В 37-й переводим измерения с лидара в формат float.

В 38-й строчке ставим ограничения от 0.3 до 5.6:

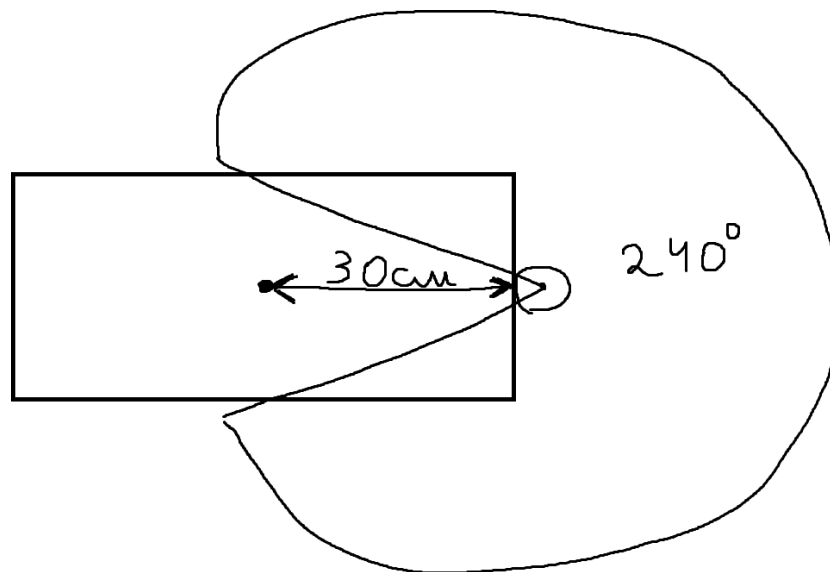


Рис.11. Ограничения лидара

Лидар находится примерно в 30-ти сантиметрах от центра робота. Если ограничение будет меньше 30-ти сантиметров, то лидар будет считывать корпус робота.

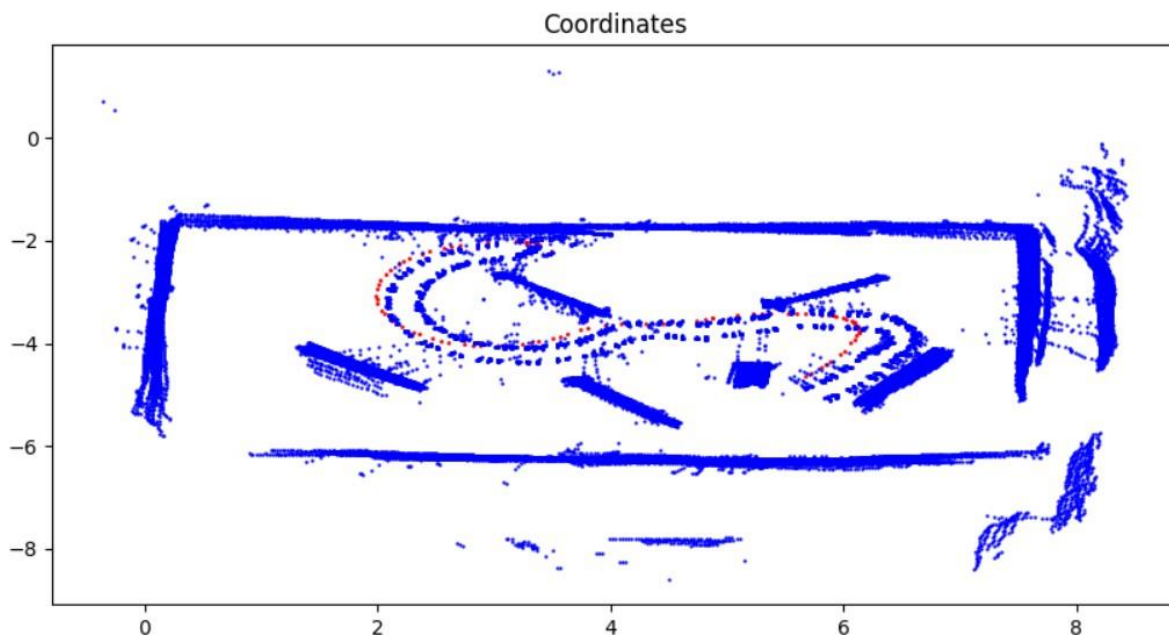


Рис.12.Пример, если ограничение стоит 10см,где лидар считывает корпус робота

В файле example10 имеется большое количество значений лидара равных 5.6. Это число означает максимальное расстояние, на которое лидар может бить лучом, либо оно значит то, что лидар не смог считать расстояние из-за того, что объект слишком близко.

39-40 строка :создание локальных координат: проекция расстояний по x и y составляющей.

К проекции x прибавляем 30 см потому, что лидар вынесен на 30 см.

41-42: Создание глобальных координат: прибавляем к координатам лидара из example10 значения локальных координат точек и прибавляем проекции 30-ти вынесенных сантиметров.

45 строка: уменьшаем phi на одно значение разрешения :

$$\frac{4.19(240 \text{ градусов в радианах})}{681(\text{количество измерений})} = 0.006152716.$$

Ошибки в построении карты:

- 1)Влияние окружающей среды(освещение,пыль,дым)
- 2)Множественные отражения луча(луч попадает в неровность на стене и много раз отражается)
- 3)Электромагнитные помехи

4) Точность измерения лидара, погрешности.