Homework 2
A69032537 Ming-Kai Liu
A69033395 Yu-An Chen
*we downscale the area from 10ft x 10ft to 8.5ft x 9ft due to spatial limitation

1. Algorithm

(1)

State $x = (x_r,\, y_r,\, \theta_r,\, x_{lm_0},\, y_{lm_0},\, \ldots,\, x_{lm_n},\, y_{lm_n})$

Measurement Vector $z_i = (r_i,\, \phi_i)$, i = landmark id

$$r_i = \sqrt{(pos_{z,\,i} + 7)^2 + (pos_{x,\,i} - 2)^2}$$

$$\phi_i = \arctan\left(\frac{pos_{x,\,i} - 2}{pos_{z,\,i} + 7}\right)$$

pos = estimated apriltag's poses

(2)

Following the [EKF-SLAM (Cyrill Stachniss)]()

$$F = \frac{\mathrm{d}}{\mathrm{d}\,(x_r,\, y_r,\, \theta_r)}\left[\left(\begin{pmatrix} x_r + (v_x \cdot dt \cdot \cos\theta - v_y \cdot dt \cdot \sin\theta) \\ y_r + (v_x \cdot dt \cdot \sin\theta + v_y \cdot dt \cdot \cos\theta) \\ \theta_r + (\omega \cdot dt) \end{pmatrix} \quad \begin{matrix} 0 \\ \\ \end{matrix} \\ \begin{matrix} 0 & & & I \end{matrix}\right)\right]$$

$$H_i = \frac{1}{q}\begin{pmatrix} -\sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 & \sqrt{q}\delta_x & \sqrt{q}\delta_y \\ \delta_y & -\delta_x & -q & -\delta_y & \delta_x \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 & 0..\,.0 & 0 & 0 & 0..\,.0 \\ 0 & 1 & 0 & 0..\,.0 & 0 & 0 & 0..\,.0 \\ 0 & 0 & 1 & 0..\,.0 & 0 & 0 & 0..\,.0 \\ 0 & 0 & 0 & 0..\,.0 & 1 & 0 & 0..\,.0 \\ 0 & 0 & 0 & 0..\,.0 & 0 & 1 & 0..\,.0 \end{pmatrix}$$

$\delta_x = (x_r - x_{lm_i}),\ \delta_y = (y_r - y_{lm_i}),\ q = \delta_x^2 + \delta_y^2$

(3)

We initialize:

$$x = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Since in the beginning, there are no landmarks detected, and the initial pose is at x = 0, y = 0, theta=0.

(4)
We set the system noise to be zero since we couldn't find an appropriate way to represent the system error. For the measurement noise, we set

$$\text{noise}_r = 0.1m$$

$$\text{noise}_\phi = 1\text{ deg}$$

(5)

- When a new landmark is detected, we initialize and add its position into state vector:

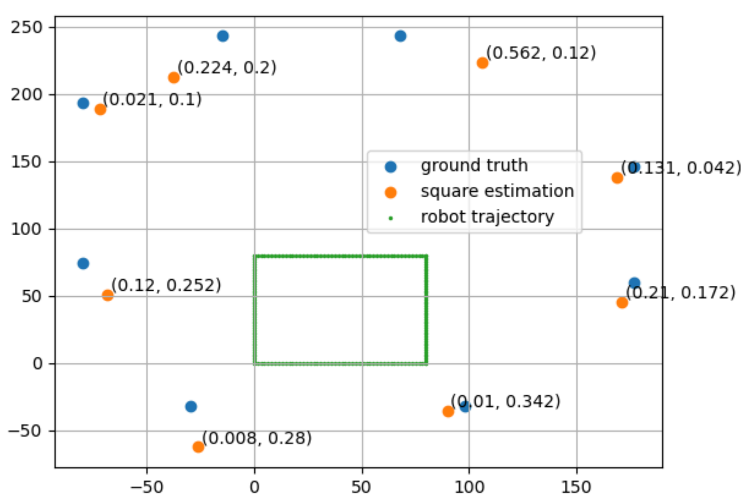
$$(x,\, y) \;=\; (r\cos{(\theta_r + \phi)},\, r\sin{(\theta_r + \phi)})$$

- When a landmark going out of view, our algorithm automatically ignore it by not update its corresponding element in **x** and **Σ**.

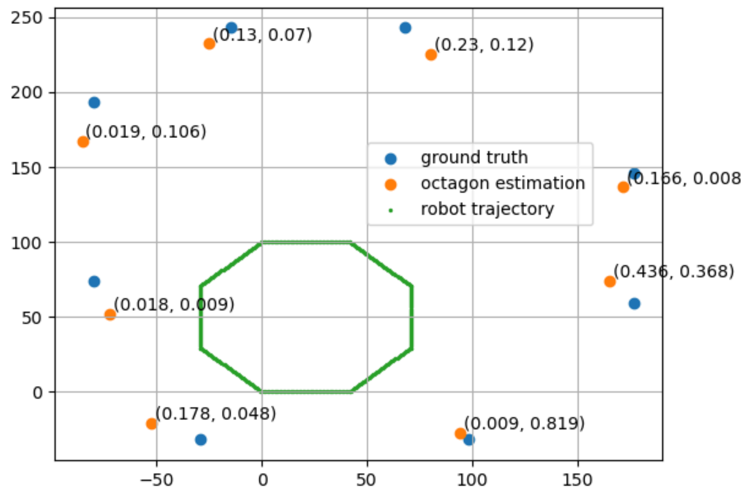
- When a previously detected landmark i reappeared, our algorithm will update its corresponding element in **x** and **Σ** by calculating corresponding H_i.

2. Result

(1)
Square: (https://youtube.com/shorts/1ZZ9lWM6Y4g?si=1LMsSj_iFafo1PI5)



Octagon: (https://youtu.be/qzQ_CbZuL54?si=Qf66jCWT8kVaIdi5)



Comment:

There is no obvious difference between the two maps. Both the estimated maps are slightly inaccurate due to the fact that our robot wasn't really running on the correct trajectory.

It can be observed that landmarks that are later discovered (landmarks on the left side) tend to be less accurate, due to the error accumulation.

Tradeoff:

In this homework, because we had no prior knowledge of the landmarks' locations, we couldn't use them to refine our estimated robot poses. Consequently, we observed that moving the robot over small ranges at times negatively impacted the algorithm's performance. This likely occurs because, with each small movement, the cumulative error increases over time.

As a trade-off, we adjusted the robot's movement to follow a point-to-point approach. While this results in the robot slightly deviating from the planned trajectory, it produces results with an acceptable level of error.

Selection of square / 8 point:

Since we set up the landmark in an 8.5ft x 9ft area, we decided to draw an octagon on a 200cm (6.5ft) square, which will ensure the camera can capture the landmark without being too close to it. We selected the square to be 80cm (2.6ft) because we found that this running distance minimizes the navigation error for our robot while still allowing it to capture and map all the landmarks.

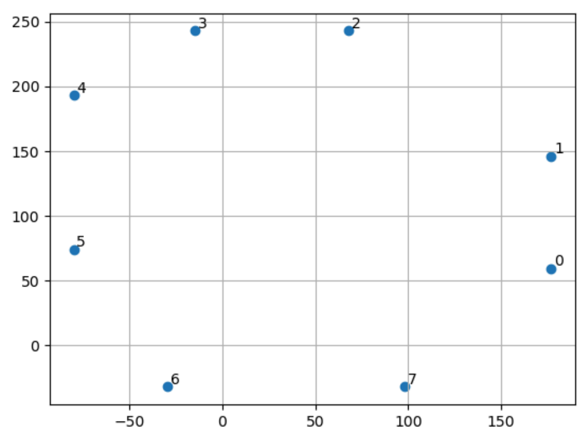
Driving multiple times:

Driving multiple times around the trajectory doesn't help our estimated map. Instead, it worsens our result. We believe this is because the longer the robot goes, the more error between its actual coordinate and its estimated coordinate (i.e. error accumulation).

Average error of landmarks:

Square: (±13.3, ±16.7) cm
Octagon: (±9.99, ±14.4) cm

(2) Environment



LM 0: (177, 59.5)
LM 1: (177, 146)
LM 2: (68, 243)
LM 3: (-14.5, 243)
LM 4: (-80, 193)
LM 5: (-80, 74)
LM 6: (-29.5, -32)
LM 7: (98, -32)