CSE 276A - Homework 3 - Report

SLAM

Wang, Shou-Yu[A69030868][a], Huang, Yao-Ting[A69030676][a]

[a] *University of California, San Diego.*

November, 2024

Abstract

In this assignment, we implement Extended KALMAN filter-based SLAM technique and evaluate its performance.

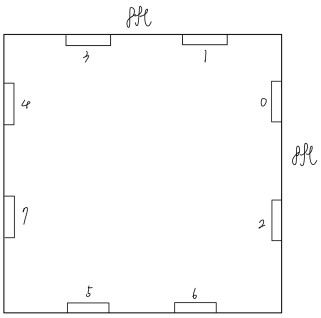

Figure 1: Environment setup.

1. Result

Check out the video of our car's navigation at link. link.

2. Environment setup

2.1. Landmarks

First we placed 8 April tags with different ids as the landmark in a 10ft × 10ft area, shown in Fig.1, and we used those landmarks to estimate and update the robot's pose. The measured positions of the landmarks shown in Table

April Tags (ID)	Position (m, m)
0	(1.45, 0.80)
1	(0.87, 2.32)
2	(1.45, 0.80)
3	(-0.13, 2.32)
4	(-0.86, 01.78)
5	(0.74, 0)
6	(0, 0)
7	(-0.86, 0.99)

Table 1: Ground truth of landmarks' positions.

3. EKF-based SLAM

3.1. Variable explanation

In the EKF-SLAM inplementation, we will maintain the state μ, measurement z. Like the following

$$\mu = \begin{bmatrix} x \\ y \\ \theta \\ m_{1,x} \\ m_{1,y} \\ \vdots \\ m_{n,x} \\ m_{n,y} \end{bmatrix}, \; z = \begin{bmatrix} l \\ \phi \end{bmatrix}$$

The states μ contains the robot's pose and the seen landmarks position. As for the measurement, we used the range-bearing representation to model the measurement.

3.2. Prediction step

When receiving a control signal u_t, we can use the Eq.1 to predict the states. The jacobian matrix of motion model shown in Eq.2. And we can thus calculate the covariance $\bar{\Sigma}$.

$$\mu_t = \mu_{t-1} + \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}^T}_{F^T} \begin{bmatrix} v_x \cos\theta\Delta_t - v_y \sin\theta\Delta_t \\ v_x \sin\theta\Delta_t + v_y \cos\theta\Delta_t \\ \omega\Delta_t \end{bmatrix} \tag{1}$$

$$G_t = \begin{bmatrix} 1 & 0 & (-v_x \sin\theta - v_y \cos\theta)\Delta_t & \cdots & 0 \\ 0 & 1 & (v_x \cos\theta - v_y \sin\theta)\Delta_t & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \tag{2}$$

$$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$$

We turn the R_t value to $0.1I$

3.3. Correction step

When the robot received a landmark detection topic, we will do the following procedures to correct the states and the covariance. We choose range-bearing representation as the measurement model $h(\mu)$, so the measurement jacobian can be calculate as shown in Eq.3

$$H_t = \frac{\partial h(\bar{\mu}_t)}{\partial \bar{\mu}_t} = \frac{1}{q}\begin{bmatrix} -\sqrt{q}\delta_x & -\sqrt{q}\delta_x & 0 & -\sqrt{q}\delta_x & -\sqrt{q}\delta_x \\ \delta_y & -\delta_x & -q & -\delta_y & \delta_x \end{bmatrix} \tag{3}$$

$$\delta = \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix} = \begin{bmatrix} \bar{\mu}_{j,x} - \bar{\mu}_{t,x} \\ \bar{\mu}_{j,y} - \bar{\mu}_{t,y} \end{bmatrix}$$

$$q = \delta^T \delta$$

$$\hat{z}_t^i = \begin{bmatrix} \sqrt{q} \\ \text{atan2}(\delta_y, \delta_x) - \bar{\mu}_{t,\theta} \end{bmatrix}$$

1. Check whether the detected landmark has been seen or nor. For each detected landmark we will examine if the landmark has been seen, go to step 3. otherwise, go to step 2.
2. We expand the covariance matrix and initialize the value $\sigma_{i,x}$ and $\sigma_{i,y}$ as 50.
3. We update the states and covariance matrix with Eq.4

$$\begin{aligned} K_t &= \bar{\Sigma}_t H_t^T (H \bar{\Sigma}_t H^T + Q_t)^{-1} \\ \mu_t &= \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t)) \\ \Sigma_t &= (I - K_t H_t)\bar{\Sigma}_t \end{aligned} \tag{4}$$

4. Discussion

4.1. Result of the map

The estimated trajectories and the position of landmarks built by different routing path shown in Fig. 2 and Fig.3.

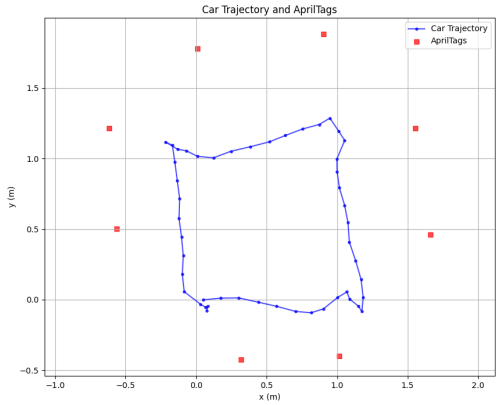

Figure 2: Map built with square trajectory.

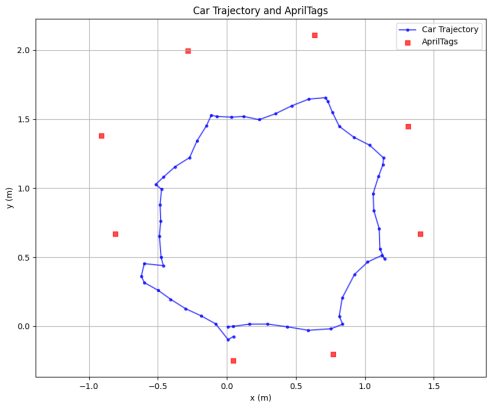

Figure 3: Map built with octagon trajectory

4.2. Performance

For square trajectory, the mean error of the landmark is 0.05 m, as for the octagon trajectory, the mean error of the landmark is 0.04 m. The map improves if we drive multiple time and it also converge with the time goes.