Extended Kalman Filter SLAM (Lab 5)

McKay Shields

November 2024

1 Introduction

Simultaneous Localization and Mapping (SLAM) is a fundamental problem in mobile robotics, involving the concurrent estimation of a robot's pose and the map of an unknown environment. This paper presents an implementation of the Extended Kalman Filter (EKF) SLAM algorithm using a robot's control inputs and lidar sensor data, with a particular focus on robust data association techniques. The nearest neighbor with double gating method was employed to reliably associate sensor measurements with existing map landmarks, mitigating issues related to sensor noise and ambiguous environments.

The algorithm was first tested on synthetic data to validate correctness and convergence under controlled conditions. Subsequently, it was applied to the Victoria Park dataset, a real-world scenario involving lidar scans from a vehicle navigating an urban park environment. The results demonstrate the EKF SLAM framework's ability to maintain accurate state estimation and map consistency, even in the presence of real-world uncertainties and data association challenges.

2 Known Data Association

2.1 Motion Model

Given a position of $[x, y, \theta]$ and a control of $[\delta_1, \text{dist}, \delta_2]$, the motion model is described by the following equations:

$$x_{n+1} = x_n + \text{dist} \cdot \cos(\theta + \delta_1)$$

$$y_{n+1} = y_n + \text{dist} \cdot \sin(\theta + \delta_1)$$

$$\theta_{n+1} = \theta_n + \delta_1 + \delta_2$$

This gives us the following matrices:

$$F_n = \frac{\partial f(\mathbf{x}_n, \mathbf{u}_n)}{\partial \mathbf{x}_n} = \begin{bmatrix} 1 & 0 & -\text{dist}\sin(\theta_n + \delta_1) \\ 0 & 1 & \text{dist}\cos(\theta_n + \delta_1) \\ 0 & 0 & 1 \end{bmatrix}$$

$$G_n = \frac{\partial \mathbf{x}_{n+1}}{\partial \mathbf{u}_n} = \begin{bmatrix} -\text{dist} \cdot \sin(\theta_n + \delta_1) & \cos(\theta_n + \delta_1) & 0 \\ \text{dist} \cdot \cos(\theta_n + \delta_1) & \sin(\theta_n + \delta_1) & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

2.2 Measurement Model

Given a position of $[x, y, \theta]$ and a landmark at $[x_\ell, y_e ll]$, the measurement model is described by the following equations:

$$\text{dist} = \sqrt{(x - x_\ell)^2 + (y - y_\ell)^2}$$

$$\delta = \arctan\left(\frac{y - y_\ell}{x - x_\ell}\right)$$

Given a position of $[x, y, \theta]$ and a measurement of $[\text{dist}, \delta]$, we can estimate the position of the landmark:

$$x_\ell = x + \text{dist} \cdot \cos(\theta + \delta)$$

$$y_\ell = y + \text{dist} \cdot \sin(\theta + \delta)$$

This gives us the following matrix:

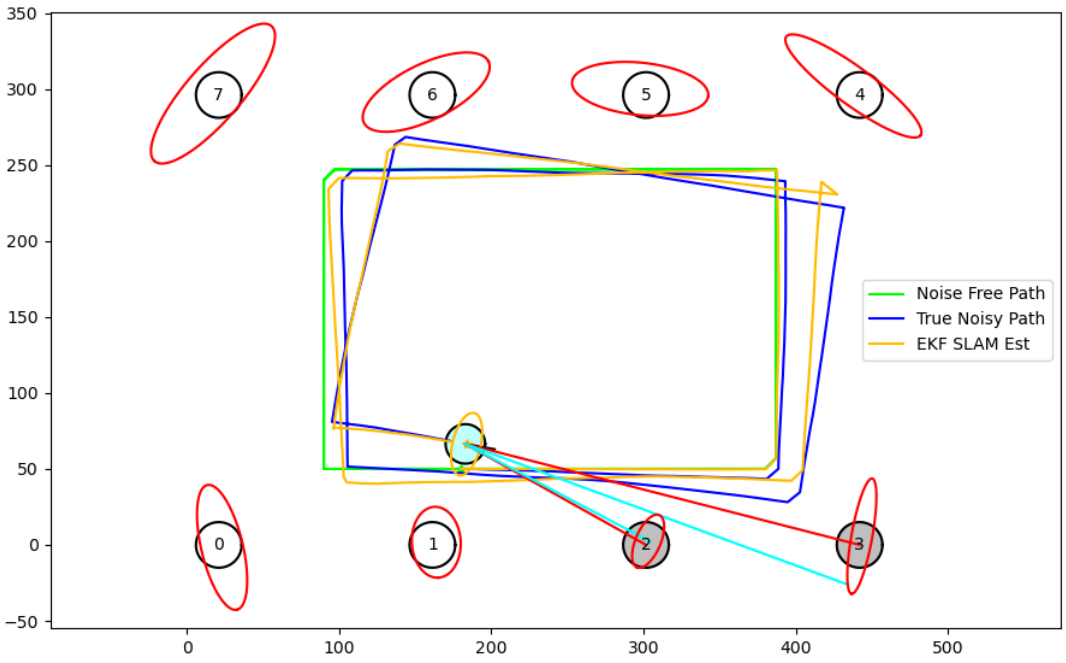
$$H_n = \begin{bmatrix} \frac{dx}{\sqrt{q}} & \frac{dy}{\sqrt{q}} & 0 \\ -\frac{dy}{q} & \frac{dx}{q} & 0 \end{bmatrix}$$

$$dx = x - x_\ell$$

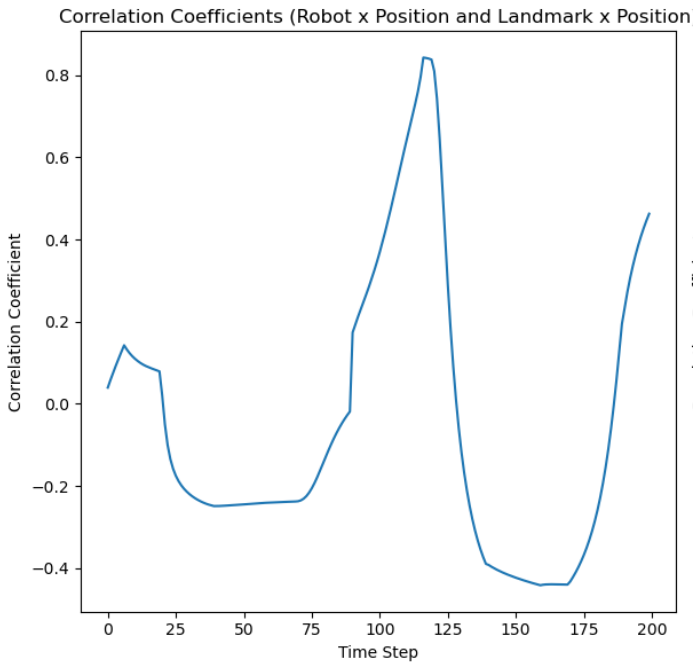
$$dy = y - y_\ell$$

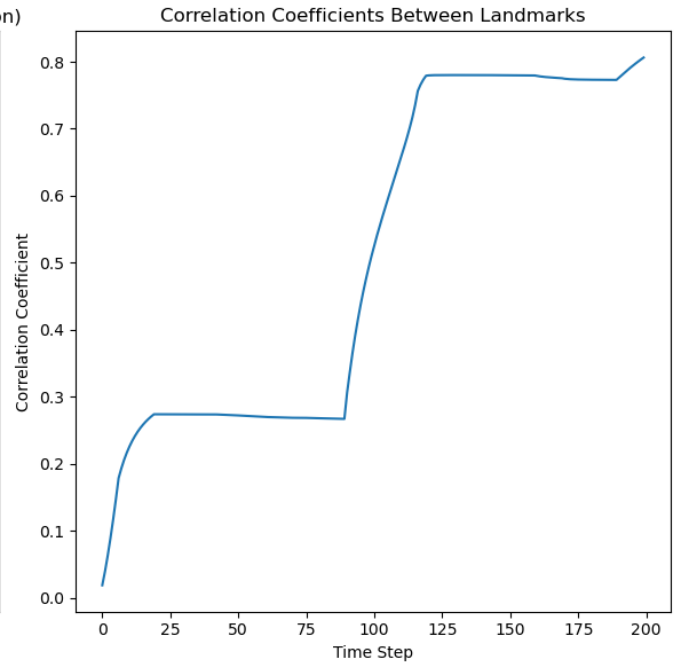
$$q = (x - x_\ell)^2 + (y - y_\ell)^2$$

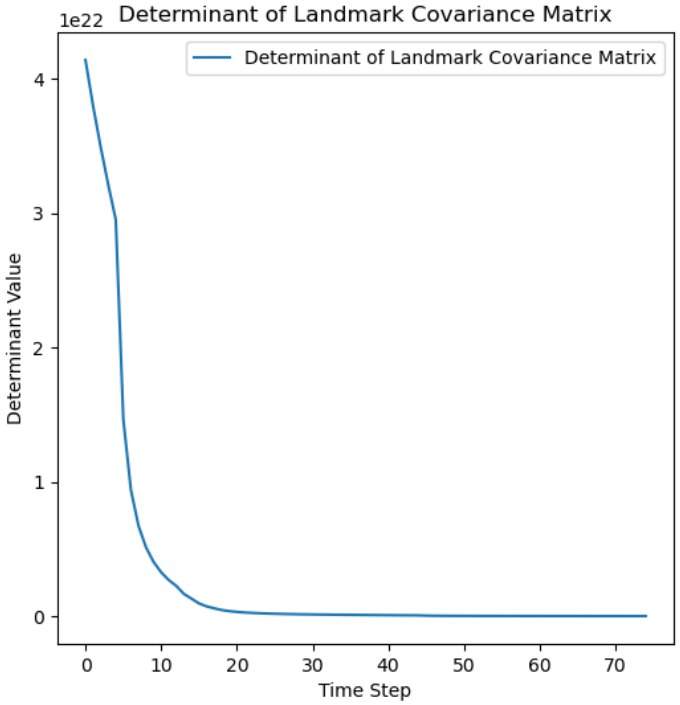
2.3 Plots of Error Ellipses



2.4 Correlation



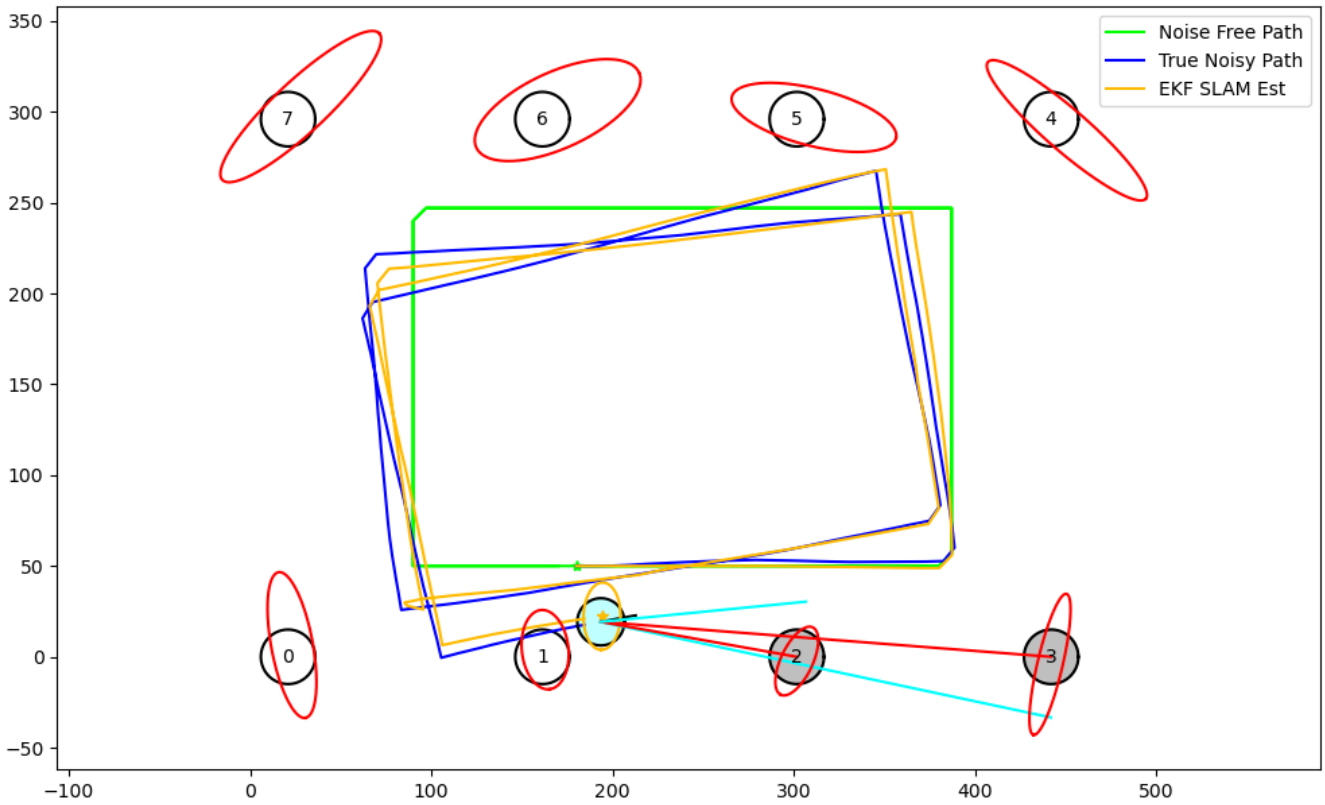




We see from these plots that the correlation does indeed increase over time. However, it is does have some jumps. The third plot shows us the determinant is also decreasing at each time step. This equates

to greater surety as we move forward along the path. The paper is justified in the claims that it makes. However, it does not include the full aspects of data association and distinguishing landmarks. Things like nearest neighbors and loop closure are an oversight in this paper. In most applications, we don't have known data association for any landmark.

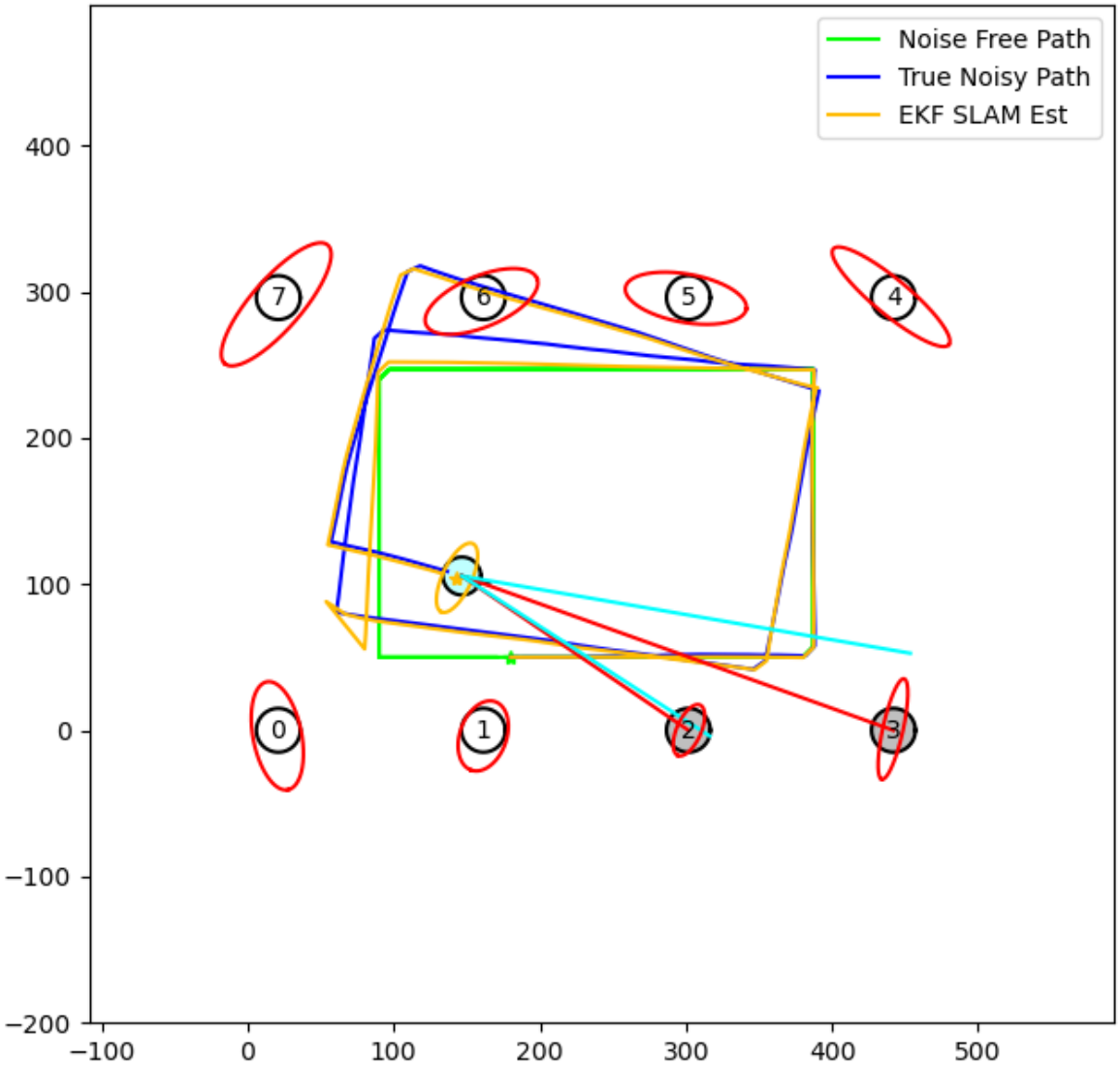
2.5 Batch Updates



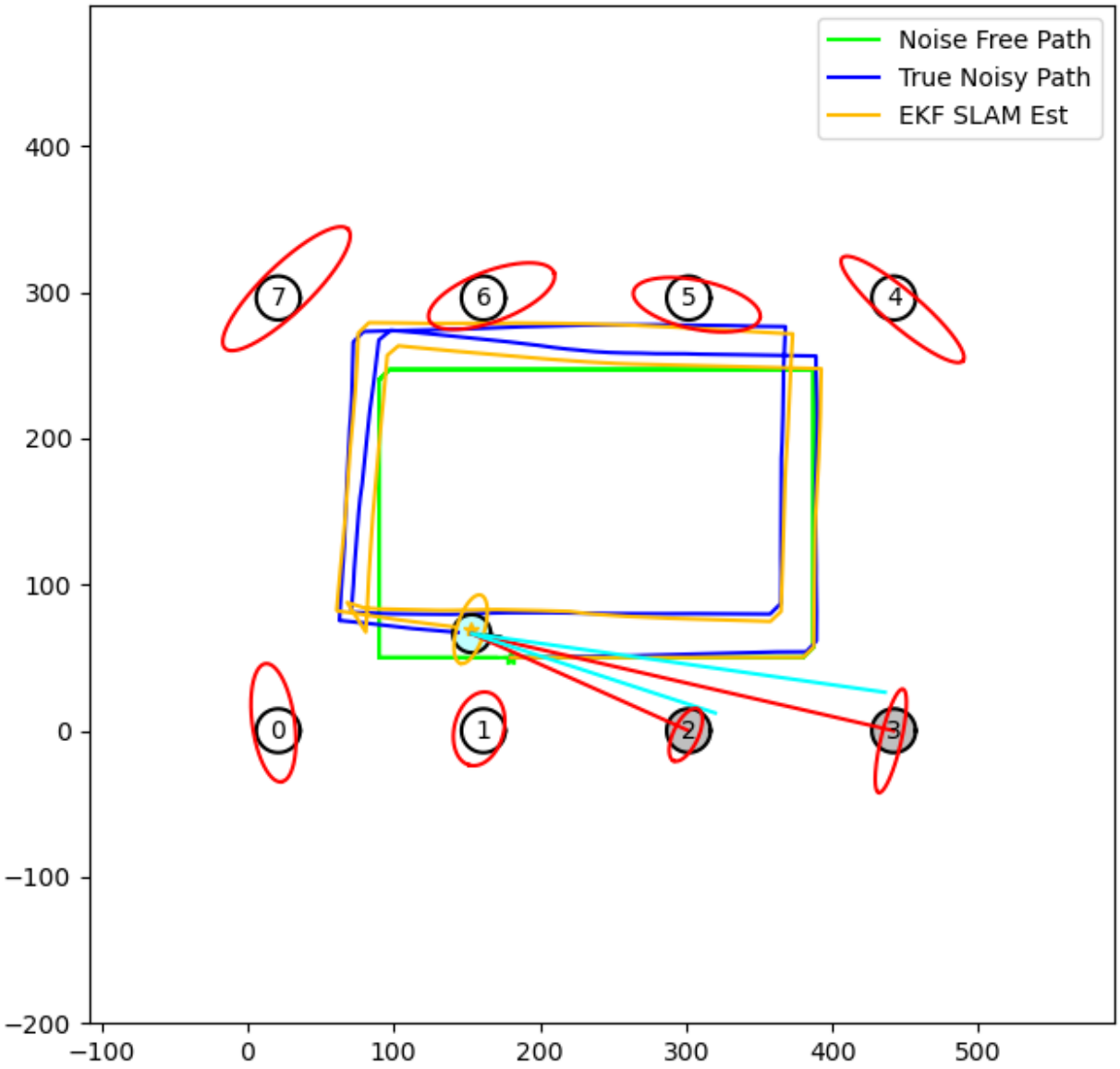
While I didn't notice extreme issues with stability, I know that the batch update was much harder to get the shaping correct. However, it comes with an advantage because the updates are much smoother especially around corners. There are no spikes are triangles in the trajectory. This is because aren't jumping just because one sensor is telling us. Rather, the update step takes into account all measurements, allowing for a gently changing path.

3 Unknown Data Association

3.1 Nearest Neighbor

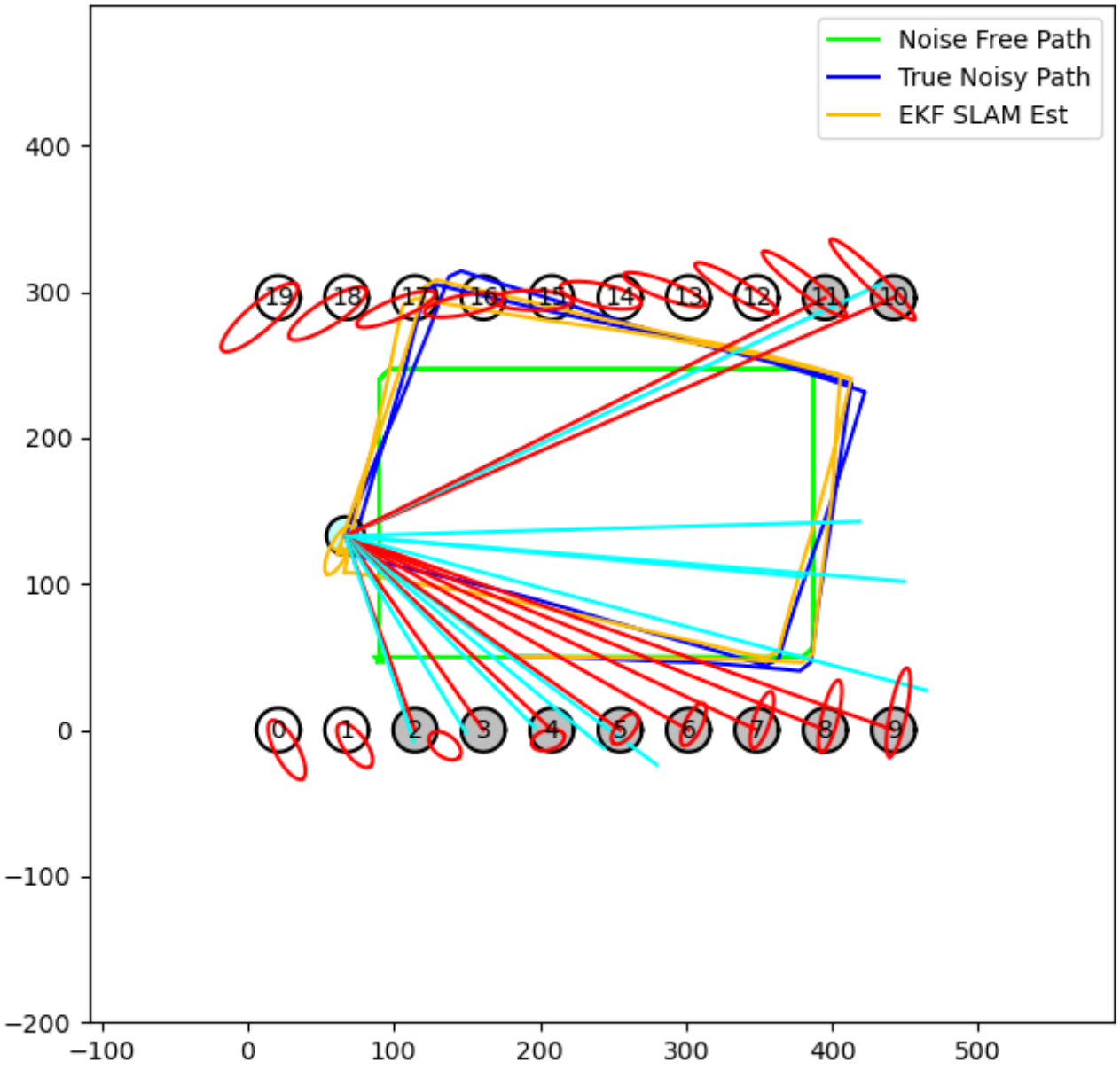


3.2 Nearest Neighbor Double Gate

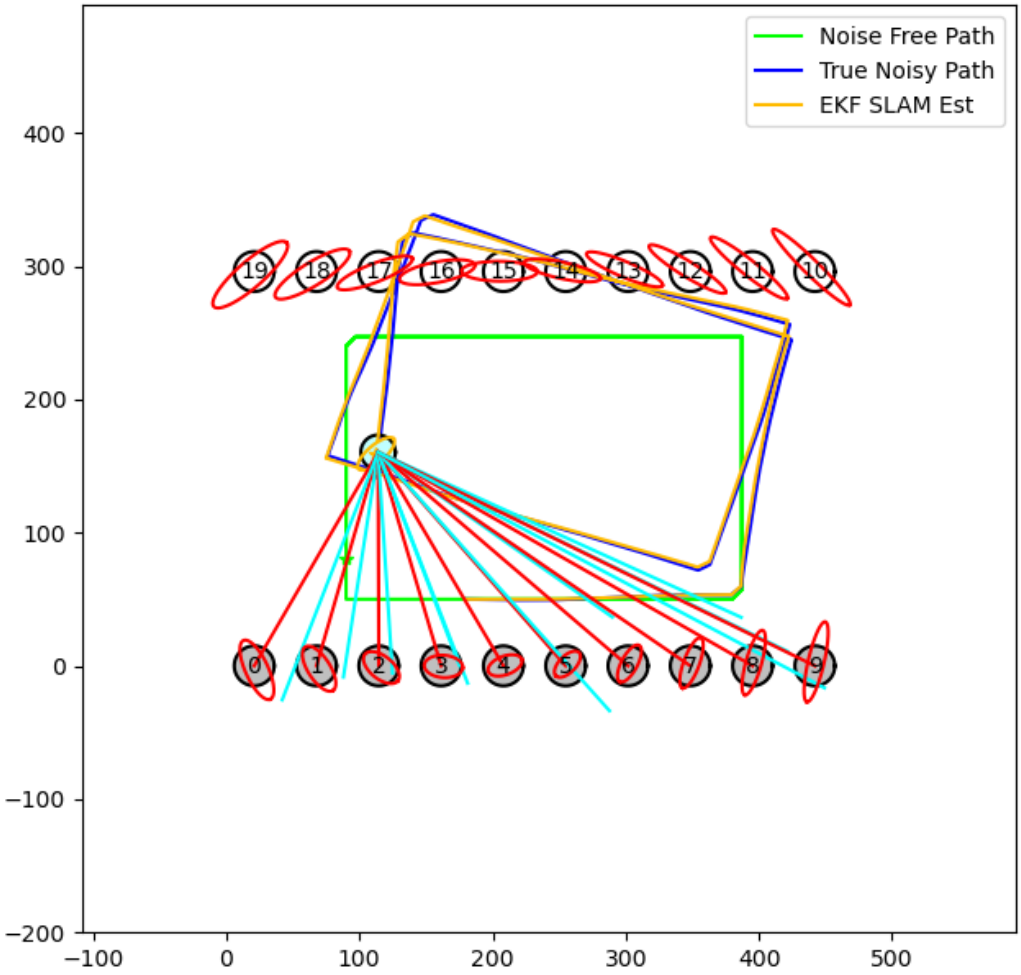


At a glance there are no distinguishing differences between the Nearest Neighbor and Nearest Neighbor Double Gate algorithms. However, watching them play out in real time, you can see that the Double Gate performed way better in real time. The Nearest Neighbor missed some of the landmarks at first and it was more uncertain. However it stabilized and looked similar to the Double Gate. The Nearest Neighbor is also very sensitive to its parameters because there is only one value that determines a new landmark, rather than 2 like in Double Gate.

3.3 Increasing Landmarks for Nearest Neighbor

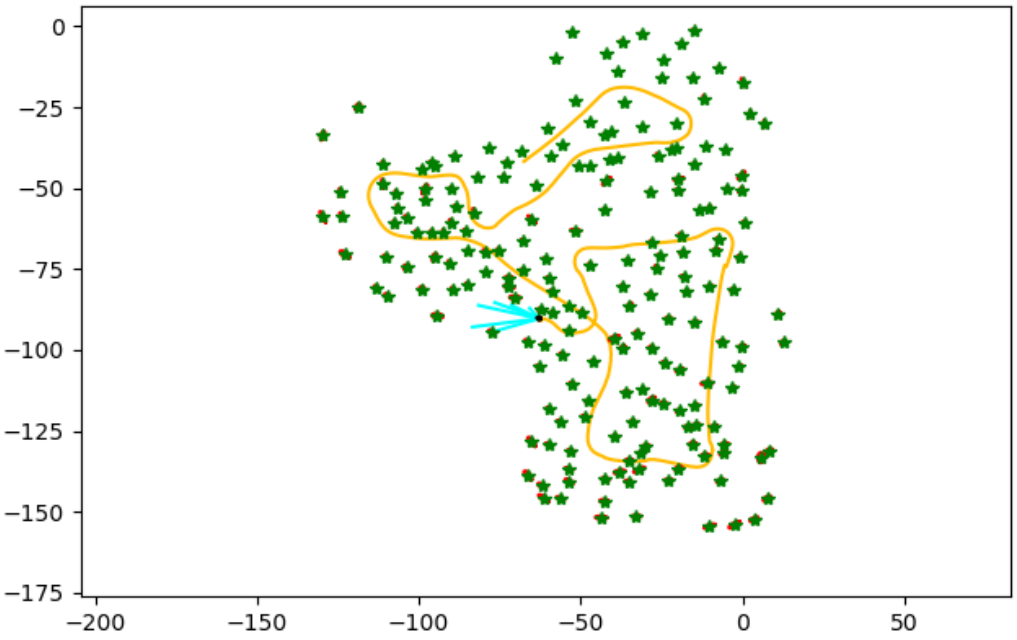


3.4 Increasing Landmarks for Nearest Neighbor Double Gate

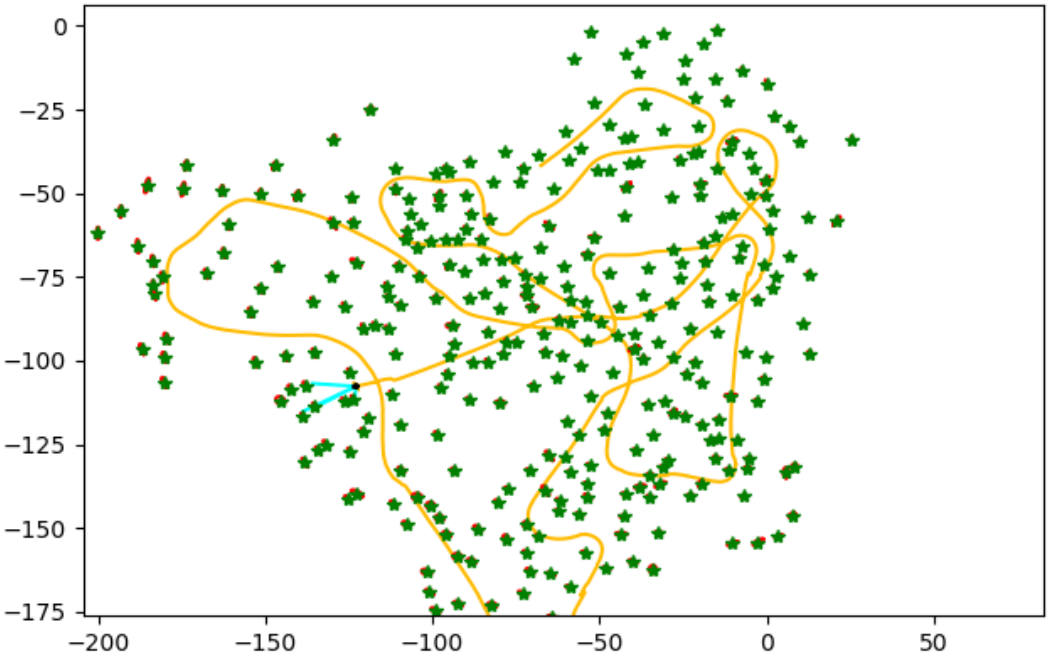


The Double Gate performed pretty much the same with more landmarks. However, you can see around landmarks 3 and 13 that Nearest Neighbor missed some landmarks. This is because it only has one opportunity to decide if measurements are coming from 1 or 2 landmarks, whereas the Double Gate has some time to pass on these measurements until it is certain if there is a new landmark or not. Error in number of landmarks can have bigger impacts on SLAM issues, especially since it can effect loop closure which unnecessarily shrinks uncertainty.

4 Victoria Park



1000 Timesteps



2000 Timesteps