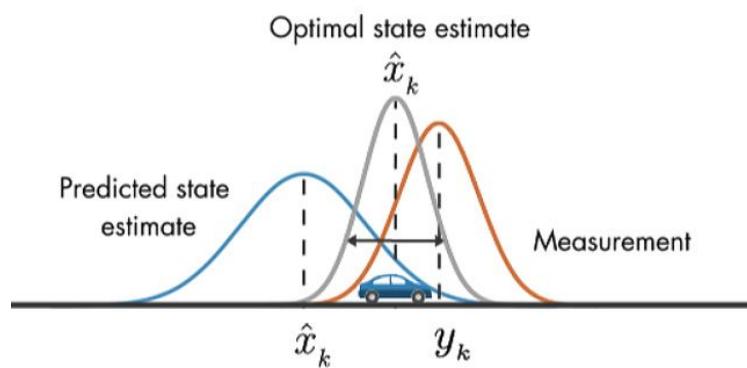


# Assignment 1

## Special Topics in Artificial Intelligence

Divyanshu Pabia  
RA2111003011373



Report & Logistics

## 1 State Estimation using Kalman Filters

### 1.1 Part a

The state is represented as  $X_t = \begin{bmatrix} x_t \\ y_t \\ z_t \\ \dot{x}_t \\ \dot{y}_t \\ \dot{z}_t \end{bmatrix}$ , actions as  $u_t = \begin{bmatrix} \delta \dot{x}_t \\ \delta \dot{y}_t \\ \delta \dot{z}_t \end{bmatrix}$  and observations as  $Z_t = \begin{bmatrix} x'_t \\ y'_t \\ z'_t \end{bmatrix}$

The motion and observational models of this problem are as follows:

$$X_{t+1} = A_t X_t + B_t u_t + \epsilon_t \quad \epsilon_t \sim N(0, R)$$

$$Z_t = C_t X_t + \delta_t \quad \delta_t \sim N(0, Q)$$

Here, the matrices  $A_t$ ,  $B_t$ ,  $C_t$ ,  $R$ ,  $Q$  are,

$$A_t = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B_t = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$C_t = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & (0.008)^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & (0.008)^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & (0.008)^2 \end{bmatrix}$$

$$Q = \begin{bmatrix} 8^2 & 0 & 0 \\ 0 & 8^2 & 0 \\ 0 & 0 & 8^2 \end{bmatrix}$$

September 18, 2023

---

Here is a plot of the actual trajectory and the observed trajectory of the vehicle, for

$$T = 300 \text{ time steps}, u_t = 0, \Delta t = 1, X_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

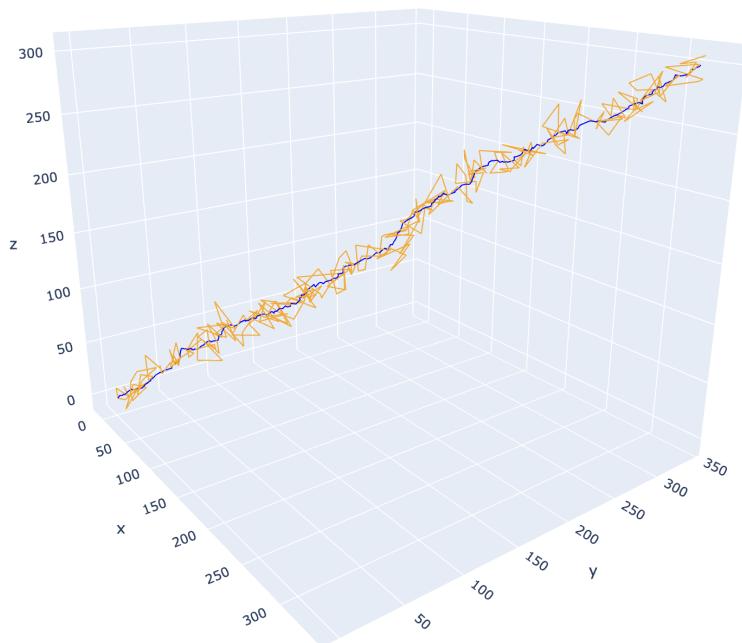


Figure 1: Observations and the actual trajectory of the plane

## 1.2 Part b

$$X_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad u_t = \begin{bmatrix} \cos(t) \\ \sin(t) \\ \sin(t) \end{bmatrix}$$

Prior belief over the vehicle's initial state is,

$$\mu_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\Sigma_0 = \begin{bmatrix} (0.008)^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & (0.008)^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & (0.008)^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & (0.008)^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & (0.008)^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & (0.008)^2 \end{bmatrix}$$

**Algorithm Kalman filter** ( $\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$ ):



```

 $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$ 
 $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$ 
 $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$ 
 $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$ 
 $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$ 
return  $\mu_t, \Sigma_t$ 

```

Figure 2: The Kalman Filter Algorithm

### 1.3 Part c

Here is a plot of the actual trajectory and the observed trajectory of the vehicle, for  $T = 300$  time steps, and the other parameters and assumptions as decided above.

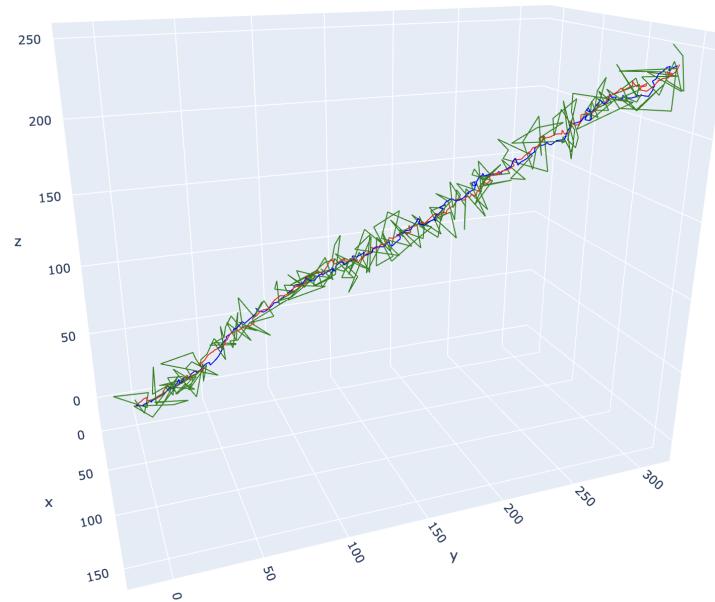


Figure 3: Actual, Estimated and Observed Trajectories in 3D

Here is a plot of the uncertainty ellipses for the projection of the estimated trajectory on the XY plane

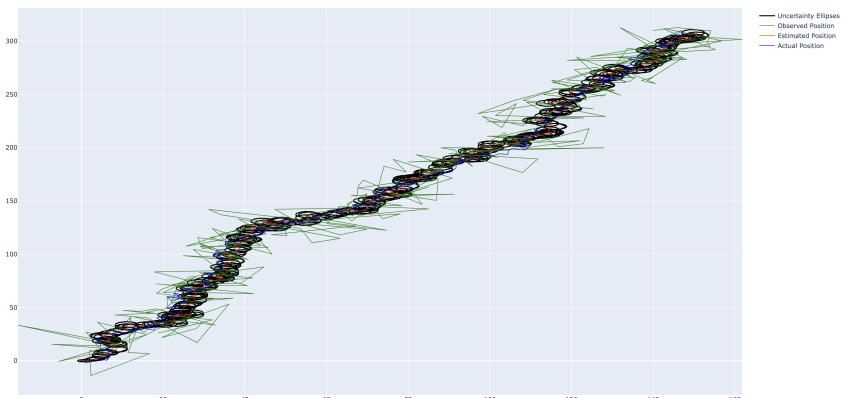


Figure 4: Uncertainty ellipses for the projection of the estimated trajectory on the XY plane

## 1.4 Part d

### a) Increase position noise

Here is a plot with position noise increased from 1 to 10:

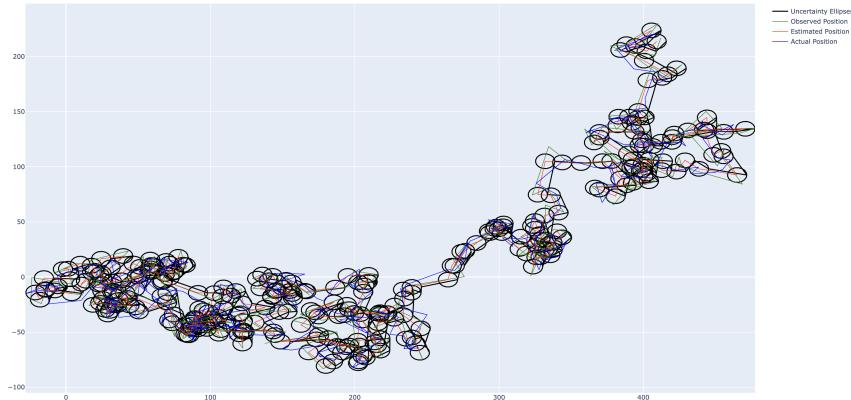


Figure 5: Uncertainty ellipses for the projection of the estimated trajectory on the XY plane

There is a lot of randomness, and this is intuitive because the position uncertainty has increased. The trajectory tends to exhibit more erratic and unpredictable behavior.

### b) Decrease position noise

Here is a plot with position noise decreased from 1 to 0.1:

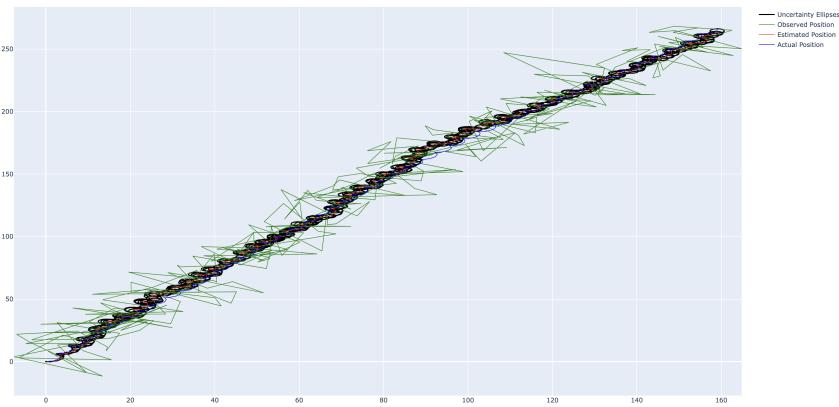


Figure 6: Uncertainty ellipses for the projection of the estimated trajectory on the XY plane

Lower noise in position updates leads to smoother trajectories that closely follow the true path. The system can estimate the object's position with greater precision, resulting in smaller and more circular uncertainty ellipses.

c) **Increase velocity noise**

Here is a plot with velocity noise increased from 0.008 to 0.08:

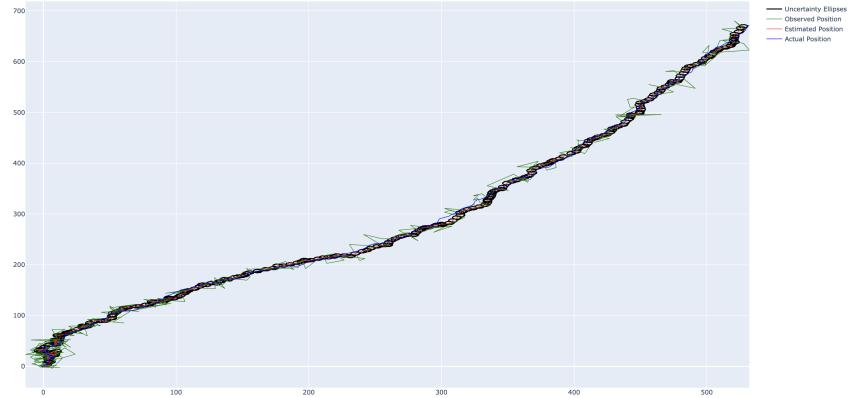


Figure 7: Uncertainty ellipses for the projection of the estimated trajectory on the XY plane

There is a lot of noise initially, and this is because the velocity uncertainty has increased, so it will take a while for it to settle.

d) **Decrease velocity noise**

Here is a plot with velocity noise decreased from 0.008 to 0.0008:

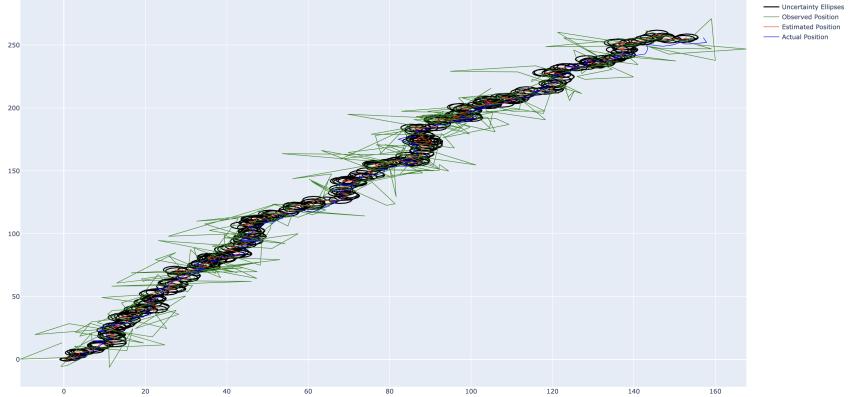


Figure 8: Uncertainty ellipses for the projection of the estimated trajectory on the XY plane

This is much like the original plot (1c).

e) **Increase sensor noise**

Here is a plot with sensor noise increased from 8 to 80:

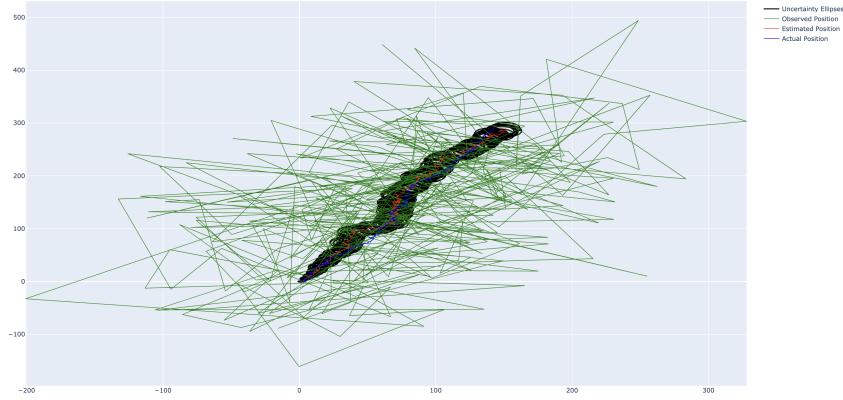


Figure 9: Uncertainty ellipses for the projection of the estimated trajectory on the XY plane

Everything has gone haywire due to the increase in sensor noise. When sensor measurements are noisy, observed trajectories tend to have more fluctuations and deviations from the true path. The system may have difficulty associating measurements with objects, leading to more frequent association errors. Uncertainty ellipses become larger, indicating greater uncertainty in position estimates.

f) **Decrease sensor noise**

Here is a plot with sensor noise decreased from 8 to 0.8:

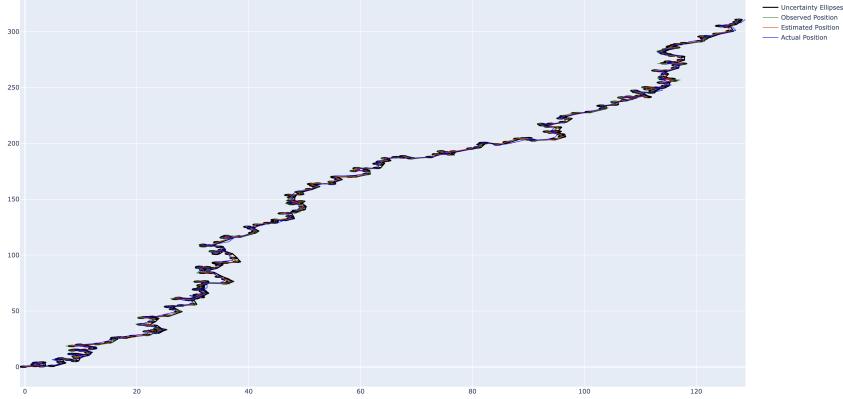


Figure 10: Uncertainty ellipses for the projection of the estimated trajectory on the XY plane

Lower sensor measurement noise leads to more stable and accurate observed trajectories that closely match the true path. The system can confidently associate measurements with objects, resulting in smaller and more precise uncertainty ellipses.

## 1.5 Part e

Assuming that the sensor observations drop out at time instants  $t = 50$  and  $t = 200$  for a period of 30 time steps and are re-acquired after that period, here is a simulation of the evolution of uncertainty in the plane's position projected on XY plane  $[xt,yt]$  by plotting the uncertainty ellipses. Here, at 2 distinct points, you can see that the sizes of the uncertainty ellipses are more than the sizes elsewhere. This is because the sensor stopped giving data and so there is an increase in the uncertainty in the position of the plane.

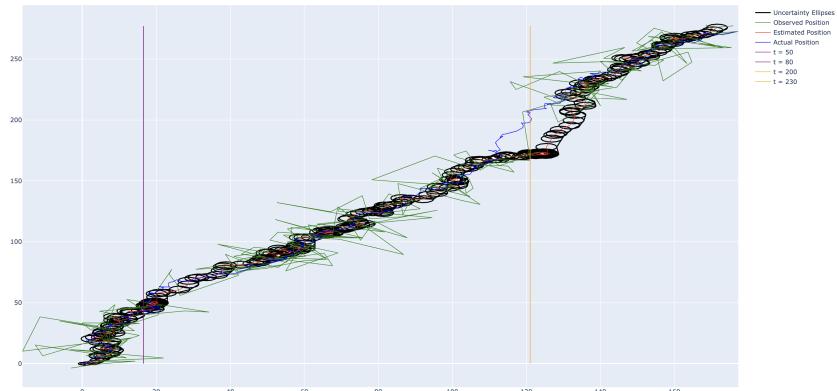


Figure 11: Uncertainty ellipses for the projection of the estimated trajectory on the XY plane

**Implementation:** In the Kalman filter algorithm, there is a prediction step and a correction step. Since we do not have the sensor values between that specific time duration, we do not correct the predictions, i.e. the beliefs are not corrected, and the predictions propagate.

## 1.6 Part f

Assuming that the X-sensor stops working at  $t = 100$  but the Y and Z sensors continue to work as usual, here is a simulation of the evolution of uncertainty in the plane's position projected on XY plane  $[xt, yt]$  by plotting the uncertainty ellipses. Here, after  $t = 100$  (Blue vertical line), you can see that the sizes of the uncertainty ellipses are increasing monotonically. This is because the X sensor stopped giving data and so there is an increase in the uncertainty in the position of the plane.

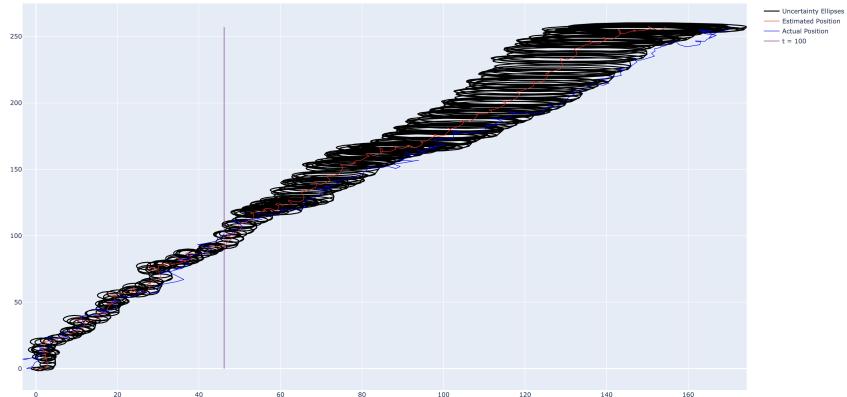


Figure 12: Uncertainty ellipses for the projection of the estimated trajectory on the XY plane

**Implementation:** In the Kalman filter algorithm, there is a prediction step and a correction step. Since we do not have the sensor value for the x-coordinate, we do not correct the prediction for the x-coordinate, everything else remains the same.

## 1.7 Part g

Here are the estimated velocities  $[\hat{x}_t, \hat{y}_t, \hat{z}_t]$  and the true velocities of the vehicle  $[\dot{x}_t, \dot{y}_t, \dot{z}_t]$ . The error in the action model for velocities is very small. Standard deviations for  $\dot{x}$  and  $\dot{y}$  are 0.008. The errors are also random with a mean of 0. So, on average, the velocity estimation is near the actual velocity.

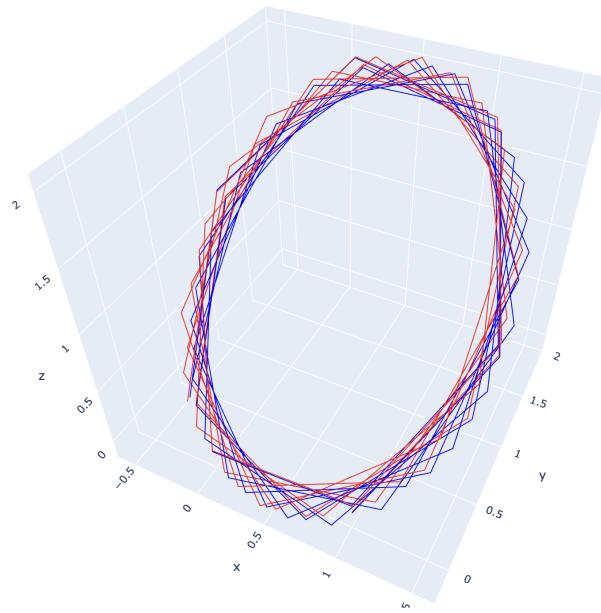


Figure 13: Plot of estimated and true velocities

## 1.8 Part h

Here is a plot showing the Data Association problem with 2 agents. I used two techniques while experimenting: Nearest Neighbour Association, and Mahalanobis Distance. In the plots of parts h and i, the graphs are generated using the Mahalanobis distance data association technique. The Mahalanobis distance takes into account the difference between the predicted state and the measurement, normalized by the covariance of the measurement. This distance is used to assess how well the object's predicted state matches the measurement.

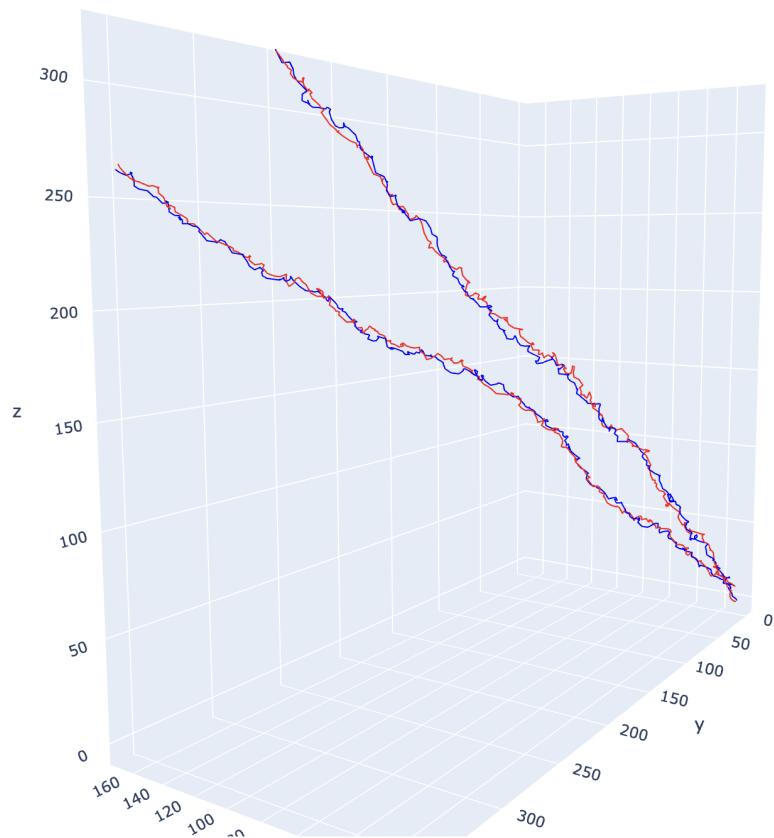


Figure 14: The estimated and actual trajectories of 2 planes with mixed observations

## 1.9 Part i

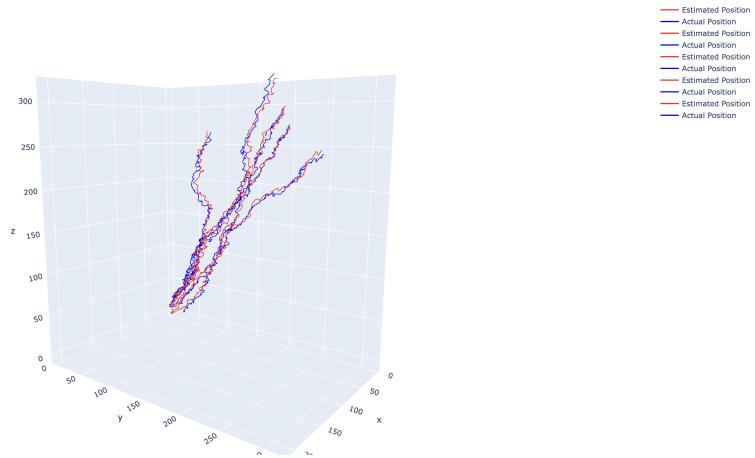


Figure 15: The estimated and actual trajectories of 5 planes with mixed observations

The algorithm is able to associate the observations with the planes correctly. This shows the correctness of the data association technique. Initial phase, the planes are close and we are not able to distinguish between the paths properly, at the boundary, the data association works correctly and when the paths diverge the data associated with them is correct. The Mahalanobis distance data association technique is robust to the number of agents. It is quite accurate too.

## 2 Question 2: Landmark Localisation

### Highlights:

- a) To account for the non-gaussian observation, we have to simulate this problem as that of an extended Kalman filter. Therefore  $X_{t+1} = f_t(X_t, u_t) + \epsilon_t$  and  $Z_t = h_t(X_t) + \delta_t$

b) Motion Model: 
$$\begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} x + \dot{x}\Delta T \\ y + \dot{y}\Delta T \\ \dot{x} \\ \dot{y} \end{bmatrix} + \epsilon \quad \epsilon \in N(0, R)$$

Observation Model: 
$$\begin{bmatrix} z_x \\ z_y \\ z_r \end{bmatrix} = \begin{bmatrix} x + \delta \\ y + \delta \\ \sqrt{(x - x_{nearest})^2 + (y - y_{nearest})^2} + \gamma \end{bmatrix} \quad \delta \in N(0, Q), \gamma \in N(0, S)$$

- c) The Jacobian for the observation model is:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{x - x_{nearest}}{\sqrt{(x - x_{nearest})^2 + (y - y_{nearest})^2}} & \frac{y - y_{nearest}}{\sqrt{(x - x_{nearest})^2 + (y - y_{nearest})^2}} & 0 & 0 \end{bmatrix}$$

- d) We do not require a Jacobian for the motion model, it is the original matrix A only.
- e) In the formula above,  $x_{nearest}$  and  $y_{nearest}$  are the coordinates of the nearest landmark within the range. When the plane is out of range of all the landmarks, we run a normal Kalman filter, not accounting for the extra observation variable. In this case, Q is a 2x2 matrix, and in the other case, it will be a 3x3 matrix.

### 2.1 Part c

Here is a zoomed-out version of the plot of my simulation.

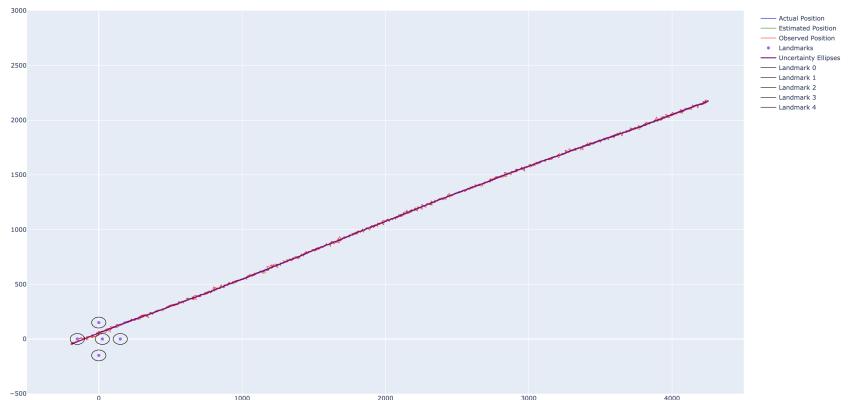


Figure 16: A zoomed out simulation of a plane for 1000 time steps

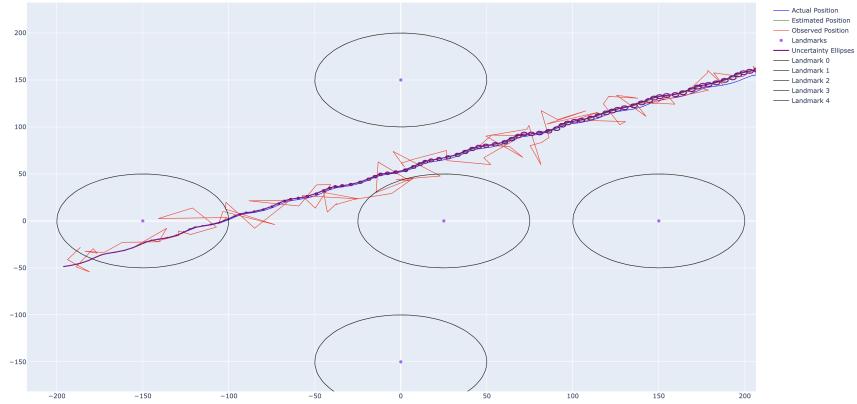


Figure 17: A zoomed in simulation of a plane for 1000 time steps

## 2.2 Part d

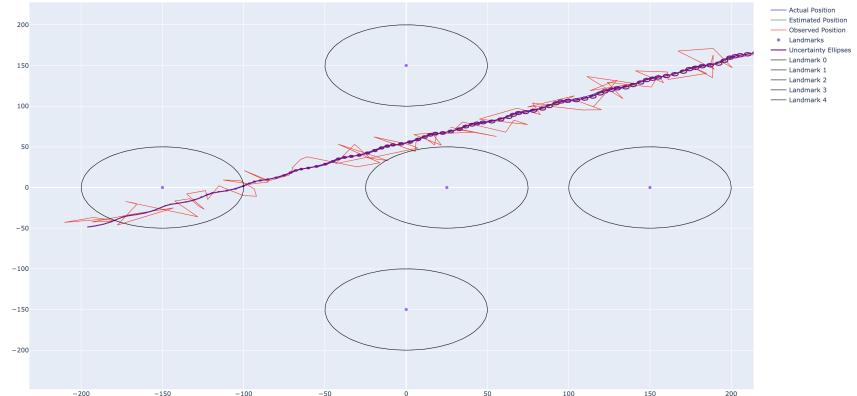


Figure 18: Increased the landmark measurement noise to 20

**Observations:** Surprisingly, there is not much of a noticeable difference when we play around this parameter from 1 to 20. Even on increasing it to 200, it remains largely the same.

### 2.3 Part e

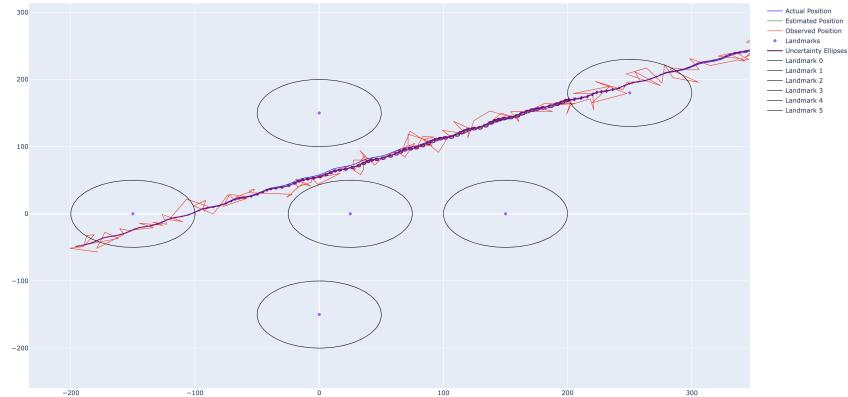


Figure 19: Added an additional landmark

**Observation:** Near the new landmark, the sizes of the uncertainty ellipses have reduced greatly. This is because the strength of the observation when it is near a landmark is quite high, therefore uncertainty decreases.