

COL778

Principles of Autonomous System

Assignment - 1

Vaibhav Seth

5th February, 2024

Contents

1 State Estimation using Kalmann Filters	2
1.1 Implementation of Motion and Observation Model	2
1.2 Kalmann Filter Implementation	3
1.3 Plots for Estimated and True Trajectory	4
1.4 Qualitative Assessment of Noise Parameters	5
1.4.1 High Noise in Position Update	5
1.4.2 Low Noise in Position Update	6
1.4.3 High Noise in Velocity Update	7
1.4.4 Low Noise in Velocity Update	8
1.4.5 High Noise in Sensor Update	9
1.4.6 Low Noise in Sensor Update	10
1.5 Radio Silence	11
1.6 Velocity Plots	12
2 State Association	15
2.1 Brute Force	15
2.1.1 Manhattan Distance	15
2.1.2 Euclidean Distance	16
2.1.3 Log Likelihood	16
2.2 Hungarian Algorithm	17
2.2.1 Manhattan Distance	17
2.2.2 Euclidean Distance	17
2.2.3 Log Likelihood	18
2.3 Mean Squared Error for different Metrics	18
3 Landmark Localization	19
3.1 Formal Description	19
3.2 EKF Description	19
3.3 Simulation 1	20
3.4 Standard Deviation Variation	21
3.4.1 $\sigma_l = 0.1$	21
3.4.2 $\sigma_l = 20$	22

Please find HTML for plots [here](#)

1 State Estimation using Kalmann Filters

In this problem our agent is an aeroplane flying in \mathbf{R}^3 and has a GPS Sensor on it which can make measurements regarding its position. It also has velocity increment based controller(action). In this problem we have defined our state X_t to be defined using position and velocity of the aeroplane, as following :

$$X_t = \begin{pmatrix} x_t \\ y_t \\ z_t \\ \dot{x}_t \\ \dot{y}_t \\ \dot{z}_t \end{pmatrix} \quad (1)$$

1.1 Implementation of Motion and Observation Model

Following the assumptions of Kalmann Filter, the motion and sensor models are Linear Gaussians and can be described as following:

$$\begin{aligned} X_{t+1} &= A_t X_t + B_t u_t + \epsilon_t \\ \epsilon_t &\sim \mathcal{N}(0, Q) \\ Z_t &= C_t X_t + \delta_t \\ \delta_t &\sim \mathcal{N}(0, R) \\ A_t &= \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} B_t = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} C_t = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \quad (2) \\ Q_t &= \begin{pmatrix} 1.44 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.44 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.44 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0001 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0001 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0001 \end{pmatrix} R_t = \begin{pmatrix} 49 & 0 & 0 \\ 0 & 49 & 0 \\ 0 & 0 & 49 \end{pmatrix} \end{aligned} \quad (3)$$

The simulations were carried out with 1 update every second, for 500 seconds, and the initial conditions were:

$$X_0 = \vec{0}$$

The following plot was observed

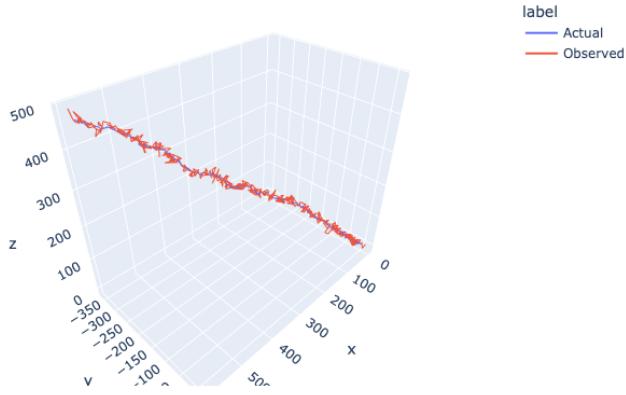


Figure 1: Plot of actual and observed trajectories

1.2 Kalmann Filter Implementation

We use the following standard algorithm for implementing Kalmann Filtering:

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

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

Action

We take the matrices as mentioned above, and initialize matrices μ_0 and Σ_0 and the initial belief becomes $X_0 \sim \mathcal{N}(\mu_0, \Sigma_0)$. At time t, belief $X_t \sim \mathcal{N}(\mu_t, \Sigma_t)$. For our simulation we've taken:

$$\mu_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (4)$$

$$\Sigma_0 = \begin{pmatrix} 0.0001 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0001 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0001 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0001 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0001 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0001 \end{pmatrix}$$

1.3 Plots for Estimated and True Trajectory

The control updates for the simulation were :

$$u_t = (\sin(t), \cos(t), \sin(t))$$

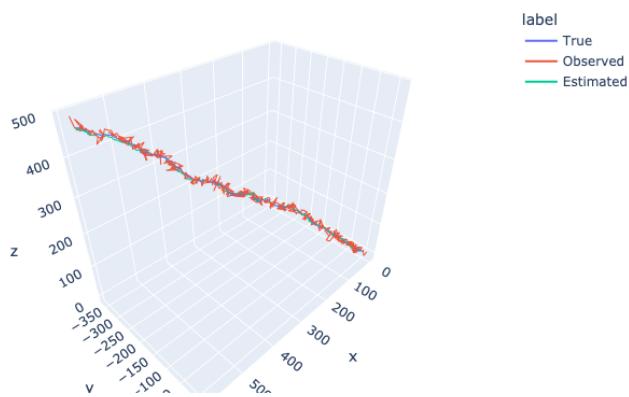


Figure 2: Plot for Estimated, Actual and Observed Trajectory

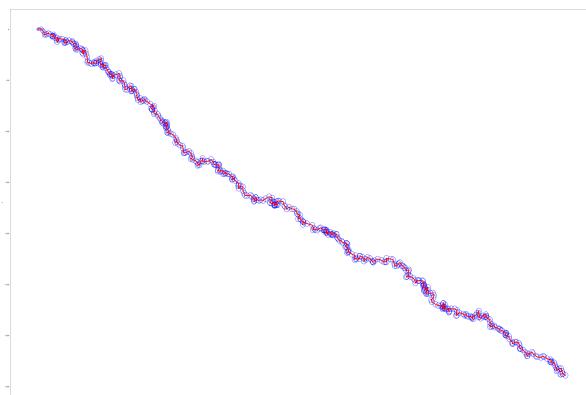


Figure 3: Uncertainty Ellipses in the estimated trajectory

1.4 Qualitative Assessment of Noise Parameters

1.4.1 High Noise in Position Update

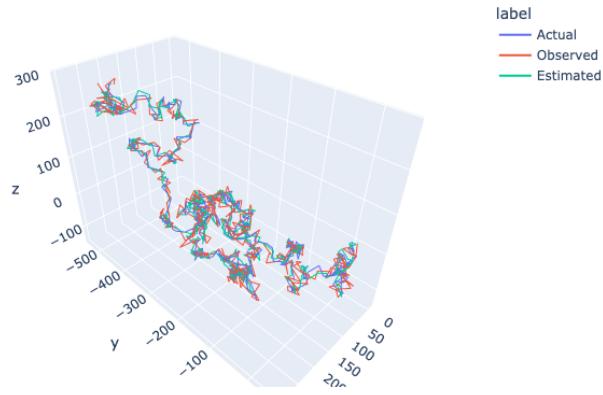


Figure 4: Plot for Estimated, Actual and Observed Trajectory

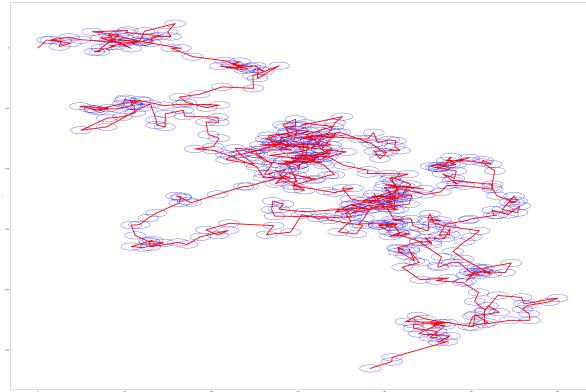


Figure 5: Uncertainty Ellipses in the estimated trajectory

We can see that due to high noise in the update parameters, the path is very haphazard. The uncertainty ellipses are also wider in this case. The predictions are not confident as the actual motion is highly unstable.

1.4.2 Low Noise in Position Update

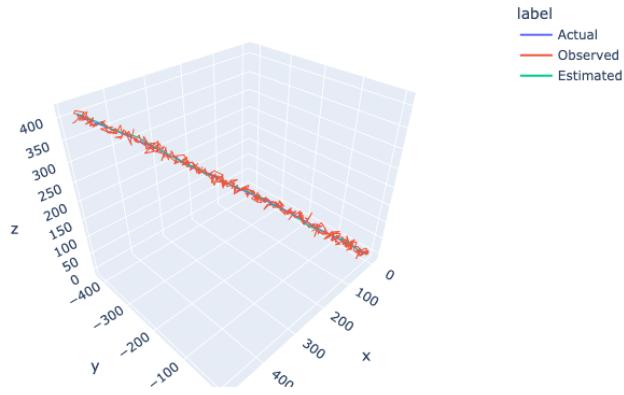


Figure 6: Plot for Estimated, Actual and Observed Trajectory

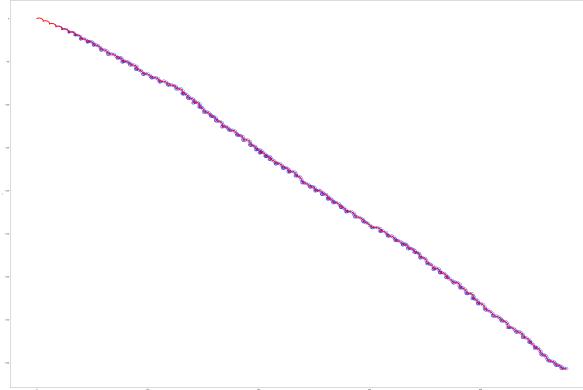


Figure 7: Uncertainty Ellipses in the estimated trajectory

We can see that due to low noise in the update parameters, the path is fairly straightforward. The uncertainty ellipses are also narrower in this case. The predictions are highly confident as the actual motion is highly stable.

1.4.3 High Noise in Velocity Update

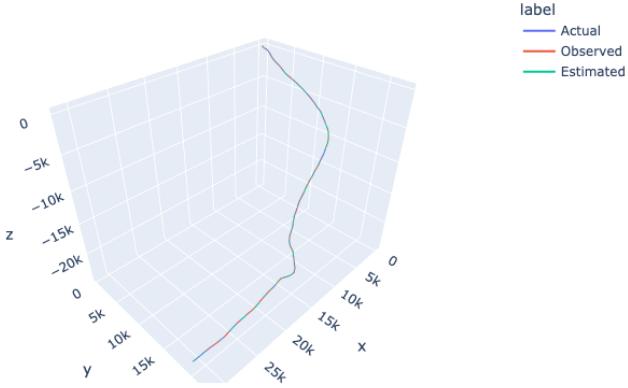


Figure 8: Plot for Estimated, Actual and Observed Trajectory

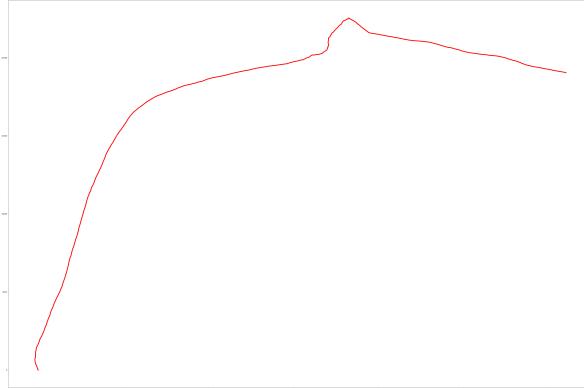


Figure 9: Uncertainty Ellipses in the estimated trajectory

We can see that due to high noise in the update parameters, the path is quite different from the previous ones. This is because of noise causing higher fluctuations in velocity update which results in a different path. As a result of varying velocities, the position updates are haphazard and position are spread far apart. Hence in Figure 9, we do not see ellipses. The following figure is a plot for only 100 time steps. We can see sparse ellipses which support our hypothesis.

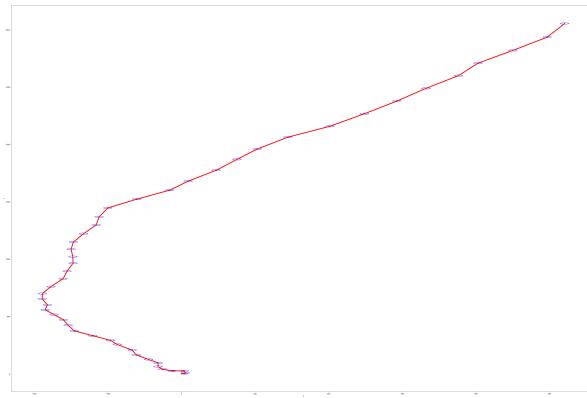


Figure 10: Uncertainty Ellipses in the estimated trajectory till time 100

1.4.4 Low Noise in Velocity Update

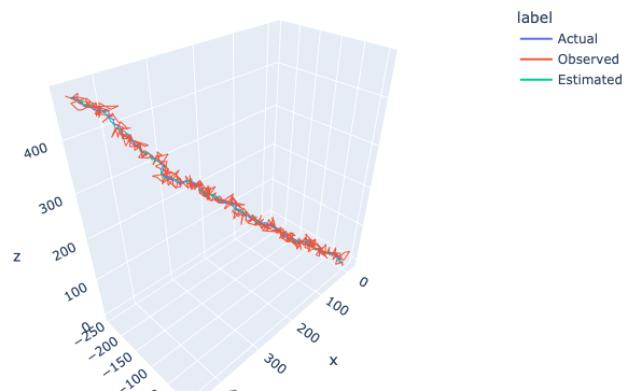


Figure 11: Plot for Estimated, Actual and Observed Trajectory

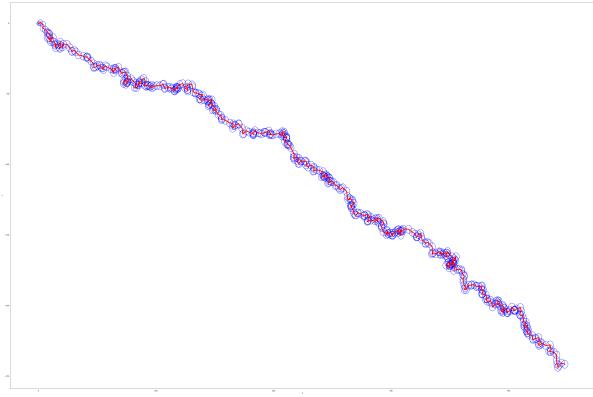


Figure 12: Uncertainty Ellipses in the estimated trajectory

The path is more or less the exact path that would have been seen if there was no noise. As a result of the reduced noise, observations are less noisy and so is the estimated path and the ellipses are also narrow.

1.4.5 High Noise in Sensor Update

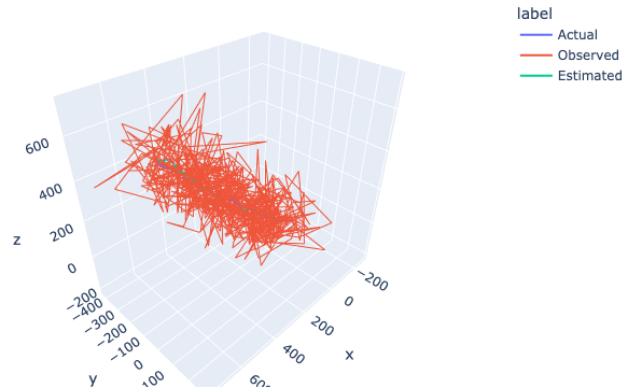


Figure 13: Plot for Estimated, Actual and Observed Trajectory

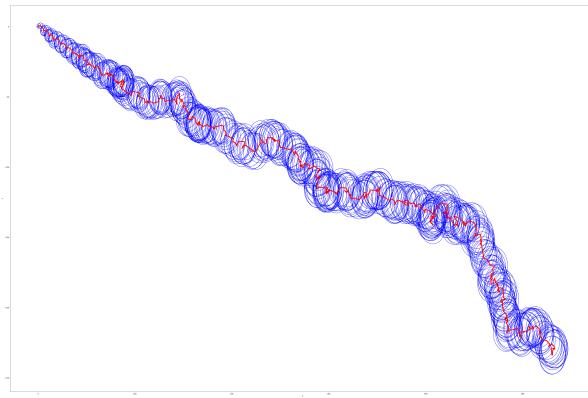


Figure 14: Uncertainty Ellipses in the estimated trajectory

We can see that due to high noise in the sensor parameters, the path is extraordinarily haphazard. Because of the noisy observations the estimated path is also very noisy. The uncertainty ellipses are very wide.

1.4.6 Low Noise in Sensor Update

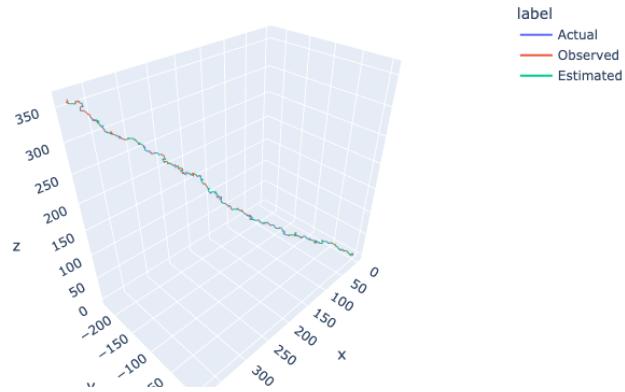


Figure 15: Plot for Estimated, Actual and Observed Trajectory

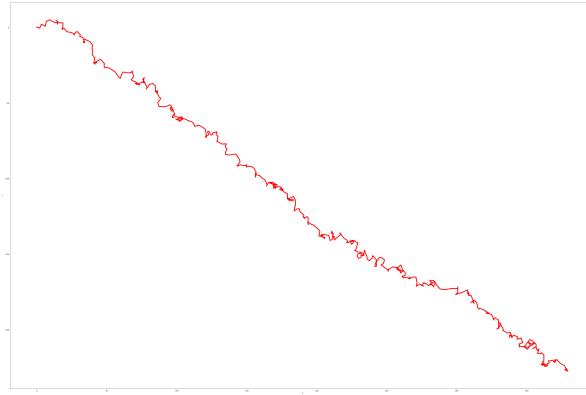


Figure 16: Uncertainty Ellipses in the estimated trajectory

We can see that due to low noise in the sensor parameters, the path is very nice. The uncertainty ellipses are much narrower in this case as observations closely follow actual, and as a result estimated follow actual.

1.5 Radio Silence

In the moments of radio silence, we propagate the belief by using the prediction update step only. The following observations are made :

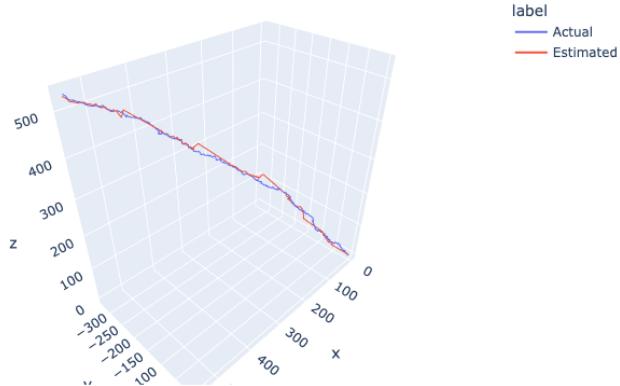


Figure 17: Plot for Estimated, Actual and Observed Trajectory

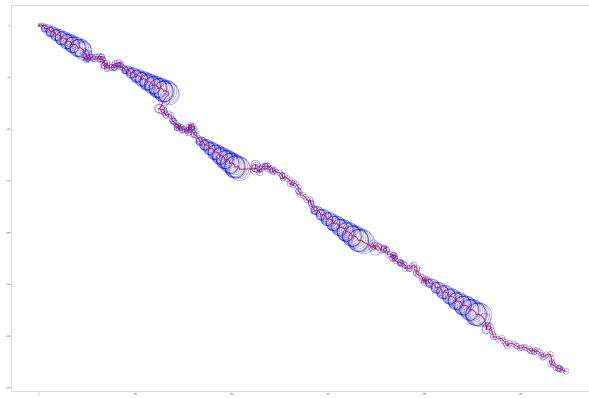


Figure 18: Uncertainty Ellipses in the estimated trajectory

We can clearly see that in moments of radio silence the uncertainty ellipses increases substantially and as soon as we get the first observation after a period of silence, the uncertainty reduces.

1.6 Velocity Plots

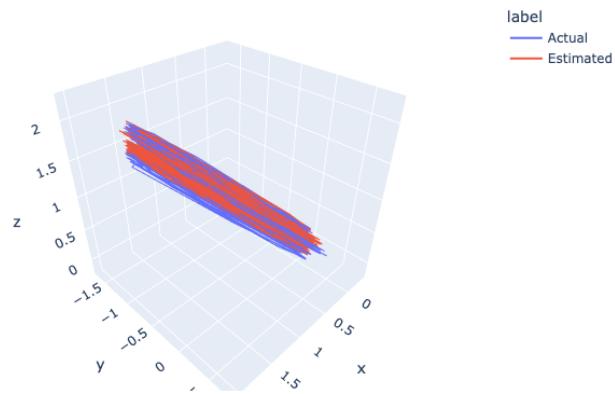


Figure 19: Plot for Estimated, Actual and Observed Trajectory when we get all observations $\mu_0 X_0$

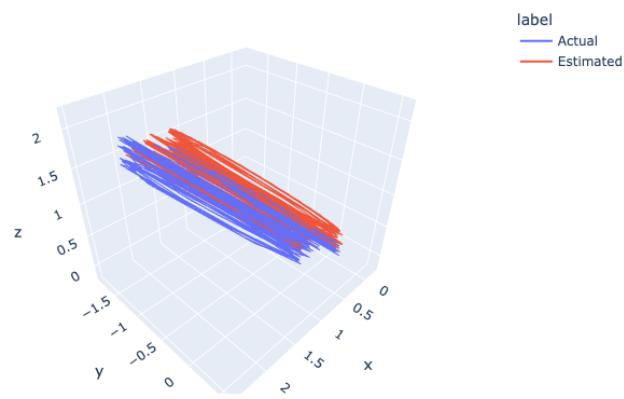


Figure 20: Plot for Estimated, Actual and Observed Trajectory with radio silence $\mu_0 X_0$

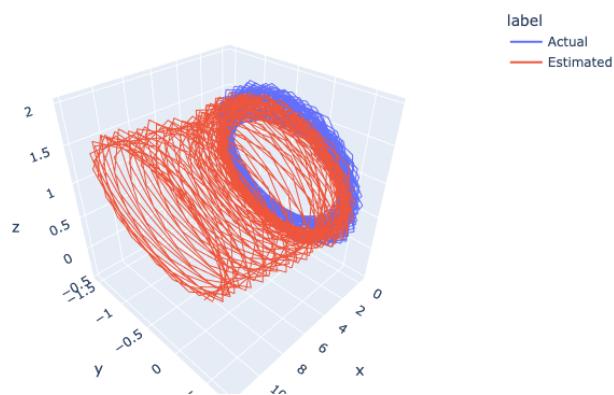


Figure 21: Plot for Estimated, Actual and Observed Trajectory with all observations $\mu_0 \neq X_0$

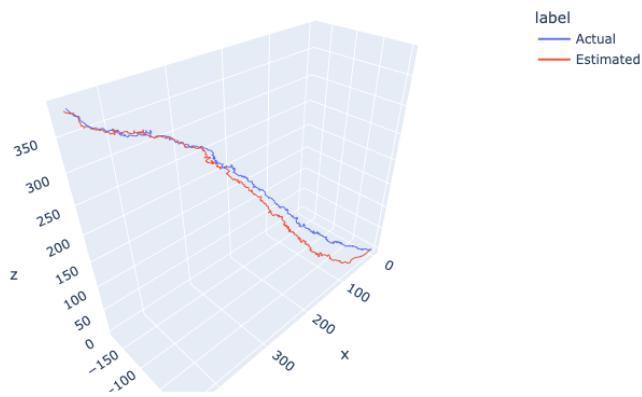


Figure 22: Plot for Estimated, Actual and Observed Trajectory $\mu_0 \neq X_0$

From the first two plots, it's not clear whether the velocities are well estimated or not. When we take a different initial belief we see that the velocity is severely terribly estimated. The estimated locations are also badly estimated. This is because we are not observing the velocities in our model and as a result the velocities do not get estimated properly and the estimated locations are also errorful.

2 State Association

The following metrics were used for performing Data Association:

- 1) Manhattan Distance : Min distance between observations and previous belief
- 2) Euclidean Distance : Min distance between observations and previous belief
- 3) Negative Log Likelihood : Max probability of points given prior belief. This also takes covariance into account

2.1 Brute Force

2.1.1 Manhattan Distance

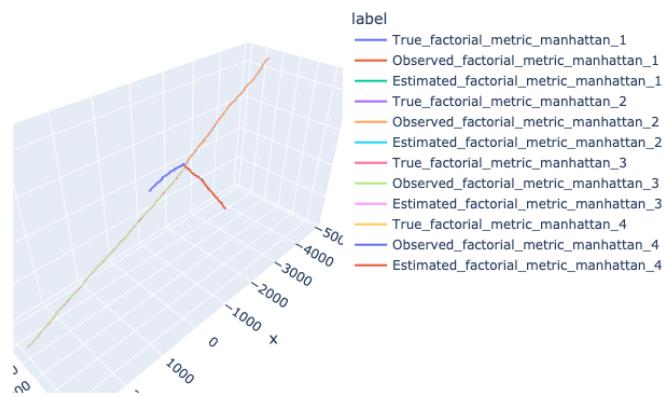


Figure 23: Plot for Estimated, Actual and Observed Trajectory for all agents

2.1.2 Euclidean Distance

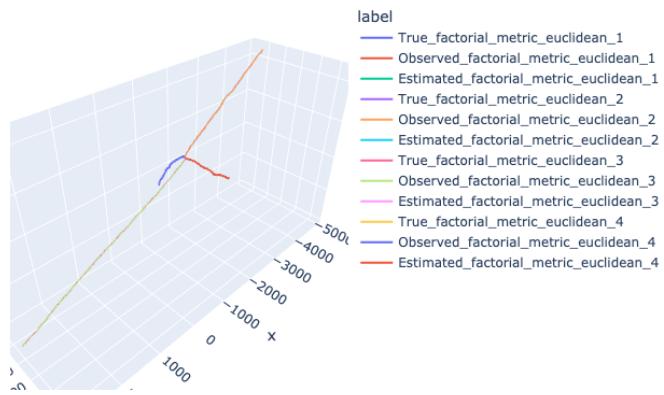


Figure 24: Plot for Estimated, Actual and Observed Trajectory for all agents

2.1.3 Log Likelihood

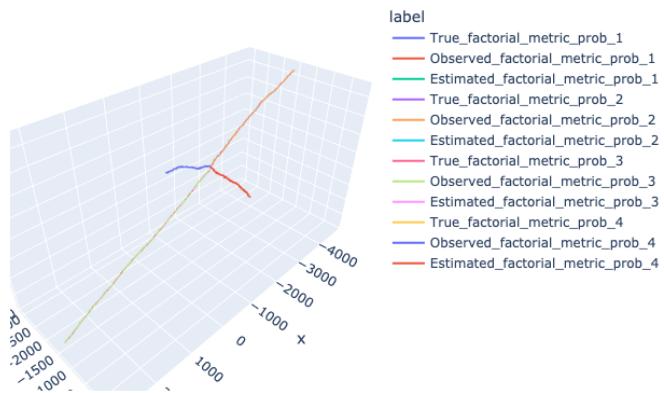


Figure 25: Plot for Estimated, Actual and Observed Trajectory for all agents

2.2 Hungarian Algorithm

2.2.1 Manhattan Distance

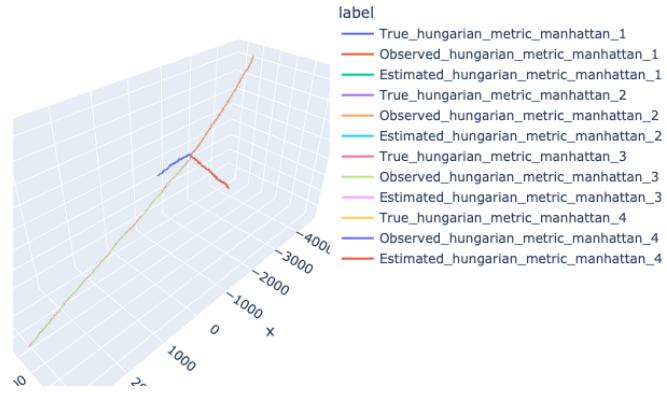


Figure 26: Plot for Estimated, Actual and Observed Trajectory for all agents

2.2.2 Euclidean Distance

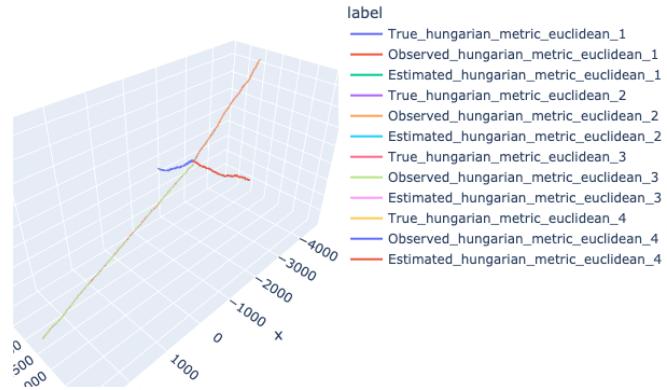


Figure 27: Plot for Estimated, Actual and Observed Trajectory for all agents

2.2.3 Log Likelihood

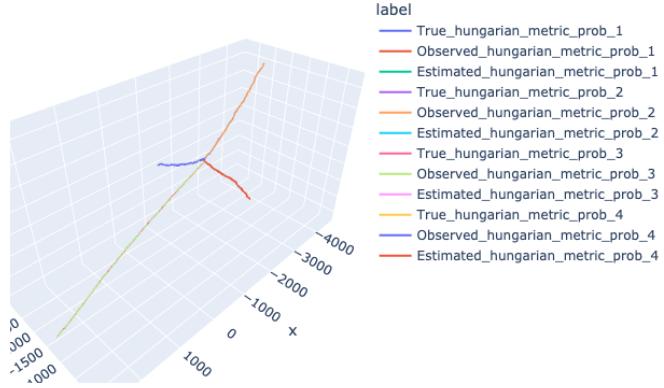


Figure 28: Plot for Estimated, Actual and Observed Trajectory for all agents

2.3 Mean Squared Error for different Metrics

Metric Name	True-Obs	True-Estimated
Hungarian Manhattan	12.1822	4.8854
Hungarian Euclidean	12.0832	4.891
Hungarian NLL	12.1522	4.7561
Factorial Manhattan	12.09299	4.8143
Factorial Euclidean	11.9823	4.8785
Factorial NLL	12.1303	4.8084

In some experiments, there were cases when the aeroplanes would pass close by and the trajectories were getting confused. This is because due to the random noise, at some instant when planes are close by, the noisy observation is better associated with some other plane. Even if planes were getting close by, our metrics were performing decently.

3 Landmark Localization

3.1 Formal Description

The observation from landmarks are incorporated as following:

$$z_t = \begin{pmatrix} x_t \\ y_t \\ z_t \\ \sqrt{(x_t - x_l)^2 + (y_t - y_l)^2 + (z_t - z_l)^2} \\ + \delta_t \end{pmatrix} \quad (5)$$

$$\delta_t \sim \mathcal{N}(0, R_l)$$

When landmark observations aren't present we use the linear Kalmann filter model.

The noise matrices are also a bit different in this case.

$$Q_t = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} * (0.01^2) \quad (6)$$

$$R_t = \begin{pmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{pmatrix}$$

$$R_l = \begin{pmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

3.2 EKF Description

Now since observation $z_t = h_t(x_t) + \delta_t$ is no longer linear we implemented Extended Kalmann Filter which requires Jacobian H_t of h_t during measurement update. Therefore we developed a Hybrid filter which does measurement updates based on the observation provided. If no landmark is involved then Kalmann Filter's measurement update is performed by estimator else EKF's update is performed

$$H(x, l) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ (x - x_l)/r & (y - y_l)/r & (z - z_l)/r & 0 & 0 & 0 \end{pmatrix} \quad (7)$$

$$r = \sqrt{(x - x_l)^2 + (y - y_l)^2 + (z - z_l)^2}$$

3.3 Simulation 1

For our simulation initial belief and starting states have been taken as describe:

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

$$\sum_0 = \begin{pmatrix} 0.0001 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0001 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0001 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0001 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0001 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0001 \end{pmatrix} \quad (8)$$

$$X_0 = \begin{pmatrix} 100 \\ 0 \\ 0 \\ 0 \\ 40 \\ 0 \end{pmatrix}$$

The control updates for the simulation are :

$$u_t = (0.128 \cos(0.032 * t), 0.128 \sin(0.032 * t), 0.01)$$

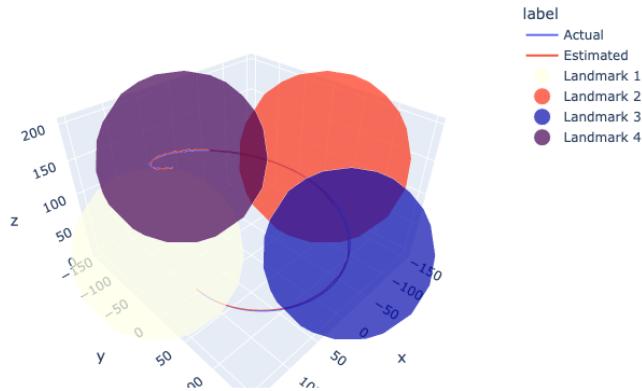


Figure 29: Plot for Estimated, Actual and Observed Trajectory with areas of influence

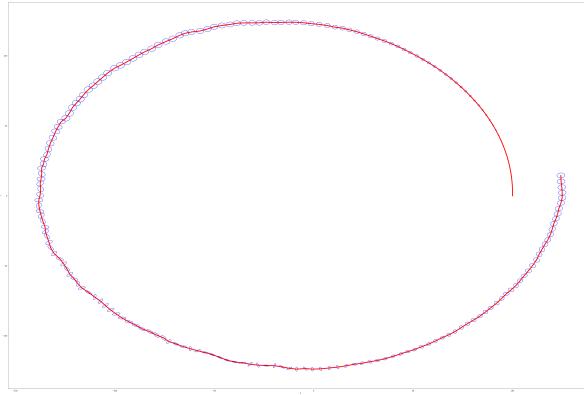


Figure 30: Plot for Uncertainty Ellipses for estimated Trajectory

We can clearly see from the above plots that in the regions of influence of landmarks, uncertainty in ellipses decrease which can be seen by narrowing of the ellipses. As the influence of landmarks end, the ellipses become prominent again.

Also, only two landmarks affect the motion for $T = 200$

3.4 Standard Deviation Variation

3.4.1 $\sigma_l = 0.1$

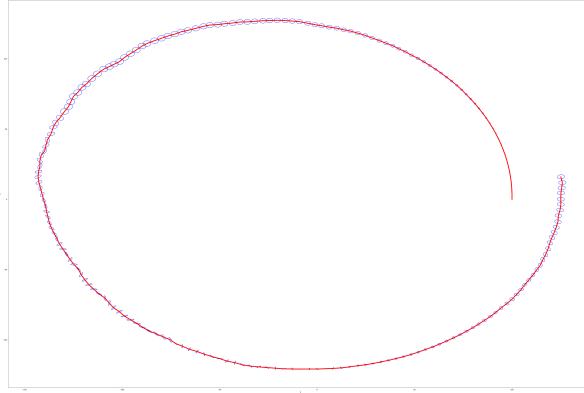


Figure 31: Plot for Uncertainty Ellipses for estimated Trajectory

We see that as a result of reduction in landmark uncertainty, the ellipses become narrower and prediction improves. In the region of landmark we see that the ellipses have almost vanished indicating good performance of the filter.

3.4.2 $\sigma_l = 20$

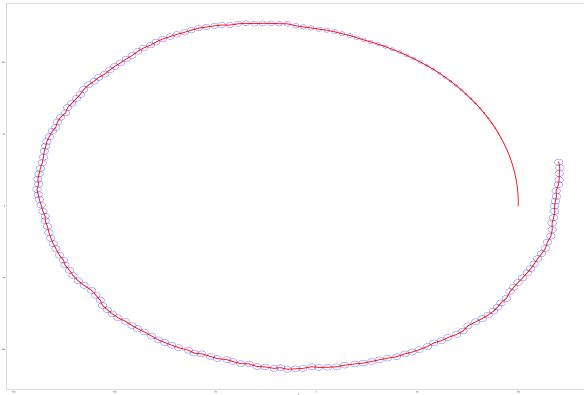


Figure 32: Plot for Uncertainty Ellipses for estimated Trajectory

In contrast to the previous case, the uncertainties have increases significantly. Even in the regions of landmark, the uncertainty ellipses are quite broad. This indicates a strong dependence on the uncertainty of observations.

The estimator might have performed better if there was no landmark, as compared to noisy landmark observations.

Please find HTML for plots [here](#) .