COL864 Assignment 1

Tamajit Banerjee | Mustafa Chasmai
2019CS10408 | 2019CS10341

February 2022

1 Kalman Filter

1.1 Motion Model

1.1.1 Model

The motion of the airplane is modelled as follows.

1. **State**

$$X_t = \begin{bmatrix} x_t \\ y_t \\ \dot{x}_t \\ \dot{y}_t \end{bmatrix} \tag{1}$$

   x_t, y_t are the coordinates of the airplane and $\dot{x}_t, \dot{y}_t$ are velocities in the x and y direction respectively.

2. **Action model**

$$U_t = \begin{bmatrix} \delta\dot{x}_t \\ \delta\dot{y}_t \end{bmatrix} \tag{2}$$

$$X_{t+1} = A_t * X_t + B_t * U_t + \epsilon_r,$$

$$A_t = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \; B_t = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \; \epsilon_r \sim N(0, R)$$

   $\delta\dot{x}_t, \delta\dot{y}_t$ represents increments provided to the velocity.

3. **Observation model**

$$Z_t = \begin{bmatrix} \delta x_t' \\ \delta y_t' \end{bmatrix} \tag{3}$$

$$Z_t = C_t * X_t + \epsilon_q$$

$$C_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \; \epsilon_q \sim N(0, Q)$$

   This refers to the position observation received from the sensors.

The model is simulated with the initial position $(0, 0)$ and v_x, v_y as 1. A gaussian with standard deviation is 0.01 is used as prior for all four state parameters. The control value is kept to $[0, 0]$.

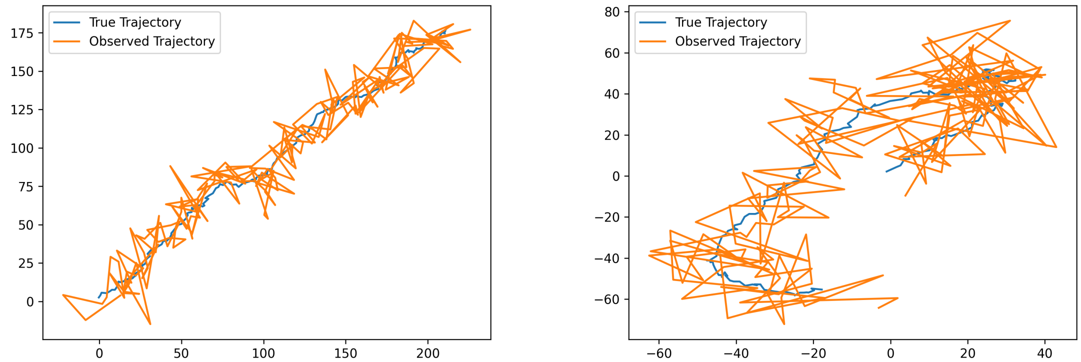

Figure 1: True and observed trajectories for the given (left) and a custom action input (right)

1.2 Kalman Filter

1.2.1 Action Step

$$\begin{aligned} \mathbf{X}_{t+1}^{(P)} &= A\mathbf{X}_t + Bu_t \\ \Sigma_{t+1}^{(P)} &= A\Sigma_t A^T + R_t \end{aligned}$$

1.2.2 Observation Step

$$\begin{aligned} K_t &= \Sigma_t^{(P)} C^T \left(C\Sigma_t^{(P)} C^T + Q_t \right)^{-1} \\ \mathbf{X}_t &= (I - K_t C)\mathbf{X}_t^{(P)} + K_t z_t \\ \Sigma_t &= (I - K_t C)\Sigma_t^{(P)} \end{aligned}$$

The following control input value is taken for simulation.

$$U_t = \begin{bmatrix} \cos(\mathrm{t}) \\ \sin(\mathrm{t}) \end{bmatrix} \tag{4}$$

Rest of the values are same as suggested in the problem statement.

1.3 Dropped Observations

On removing the sensor for timesteps $t = 10 \rightarrow 30$ and $t = 60 \rightarrow 80$, we get the following results. The implementation is done by breaking down the kalman filter algorithm into action step and observation step and thereafter applying the observation step only at desired timesteps.

Observe that the standard deviations are larger for time steps at which observation data is unavailable. This is reasonable since we do not perform the correction step in the Kalman Filter algorithm at those timesteps. Thus, the decrease in variance caused due to it is not done.

1.4 Uncertainty estimates

Observe that despite significant variance in observations, the model is able to estimate the actual state quite efficiently

.

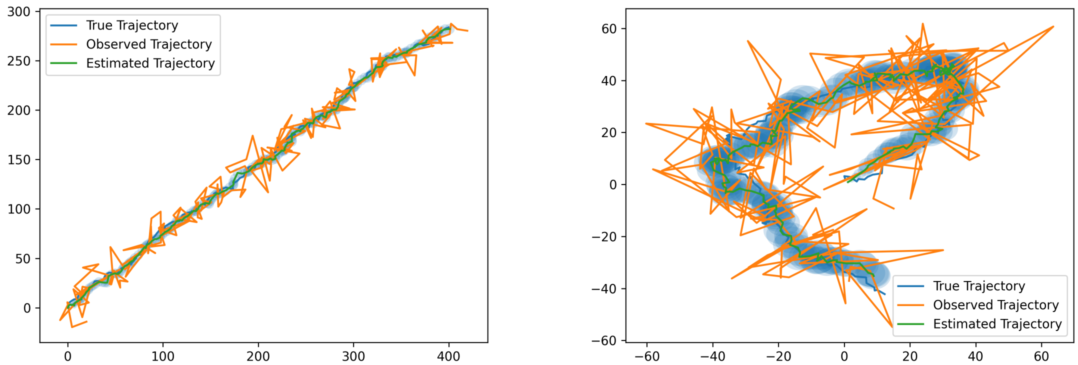

Figure 2: True, observed and estimated trajectories along with uncertainty ellipses for the given (left) and a custom attack input (right)

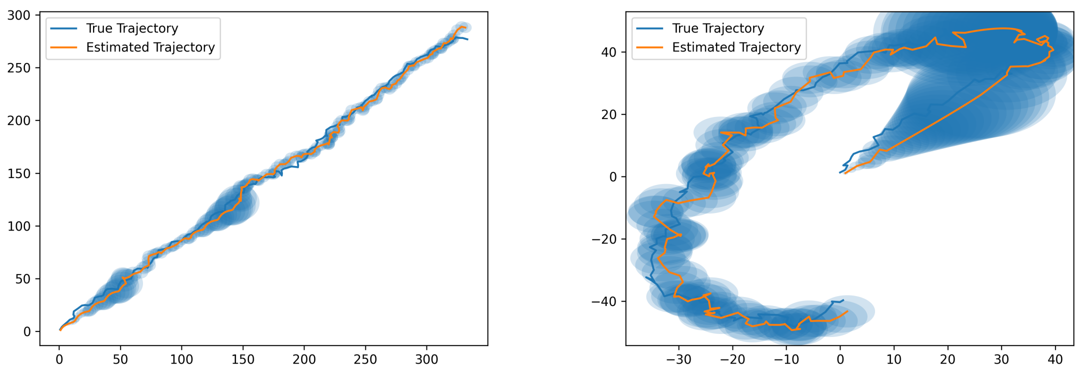

Figure 3: True and estimated trajectories along with uncertainty ellipses for the given (left) and a custom attack input (right)

1.5 Uncertainty estimates in the velocity

Even though, the velocity is not observed and despite significant variance in the velocity. We find that the the model is able to estimate the actual velocity state quite efficiently. The model is able to give a good estimate of velocity as it is the change in position per unit time which it is able to predict well.

1.6 Data association problem

The multiple target tracking problem is encountered in many situations, whenever sensor data is available from one or more wide angle of view sensors (such as radar, electronic support measures (ESM), infra-red (IR), video, etc.), providing information on the position (such as range, azimuth, elevation etc.) of "targets"' (aircraft, vessels, missiles, ground vehicles etc.). The sensors may also provide information useful for target identification purposes (size, luminosity, transmitter frequency, etc.).
These target tracking sensors typically scan their field of view at regular intervals. There may be many contacts from such a scan, from the different objects in the sensors field of view, as well as some which may be just noise. The sensor measurements from all contacts within a scan are generally processed

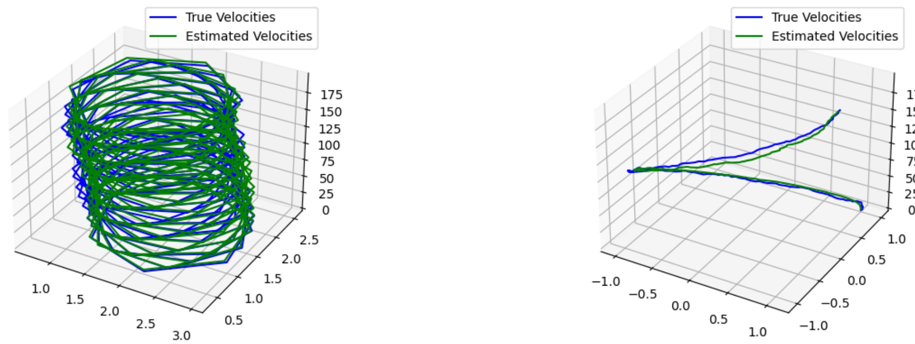

Figure 4: True and estimated velocity for the given (left) and a custom attack input (right). The z axis corresponds to time while the x and y axes are vx and vy respectively.

together, since they can safely be assumed to correspond to different objects (something which cannot be said of returns from different scans or from different sensors).
The basic tracking problem is to estimate the position and velocity of the target(s), using the available sensor data (from a sequence of scans). It is also normally desirable to be able to predict the targets' location some time in the near future. This is a problem because the sensor data normally has errors and/or ambiguities and the targets generally move between scans.
Therefore, data association problem is an important problem.
Here, we will assume that the number of measurements given by the sensor is equal to that of the number of agents where each measurement belongs to a different agent but the position measurements can be noisy. Our goal here will be to associate each of the measurements to one of the agents depending on their previous trajectories.

1.6.1 Approach

Here, we have implemented a greedy approach.
The problem setting is that there are n airplanes with estimates (μ_t, Σ_t) after incorporating action at any time step. Suppose the observations at the same time step are Z_t.

In radar tracking system, association gate is defined around a predicted target position, which accumulates uncertainty of prediction, resulting from tracking filter parametrization and past performance of the tracking process. When the following relation of norm of residual error d^2 :

$$\mathbf{S}_t = C\Sigma_t^{(P)}C^T + Q_t$$
$$\mathbf{Y}_t = C * X_t$$
$$\mathbf{d}^2 = Y_t^T S_t^{-1} Y_t$$

In the literature the above equation is known as statistical distance or Mahalanobis distance, and is a typical metric used in gating algorithms.
For a plane i, we find the probability $P_{i,Z_{jt}}$ that a measurement Z_{jt} is from it at time t, by assuming the measurement is a Normal Gaussian distribution with mean $C * \mu_t$ and covariance S_t.
Observe that these observations are shuffled randomly and we need to assign each of them to an estimate. The objective is:

$$\texttt{argmax}_p\{\Pi_{i=1}^n P_{i,Z_{p(i)t}}\}$$
where p is a permuation of $\{1, 2, ..n\}$

Maximising $\Pi_{i=1}^{n} \Pr(Z_{p(i)}|X_i)$ can be reframed as maximising the sum of log-likelihood corresponding to each estimate.
We can compute optimal p by iterating through each permutation and computing the sum of log-likelihoods. This has $O(n.n!)$ time complexity and this works well for small number of objects to track. In our implementation n is at most 4 or 5 hence the brute force method works well.
Here, we instead of taking only the optimal permutation, we also store the best 10 estimates. The intuition behind it is that if 2 or more planes come very close by or the observations are very noisy, it should not be the case that maximising the product of the probability might not be optimal. Hence we keep the top 10 assignments or estimates each time so that even if the optimal one is not the correct assignment, the algorithm is robust enough to restore the balance soon.

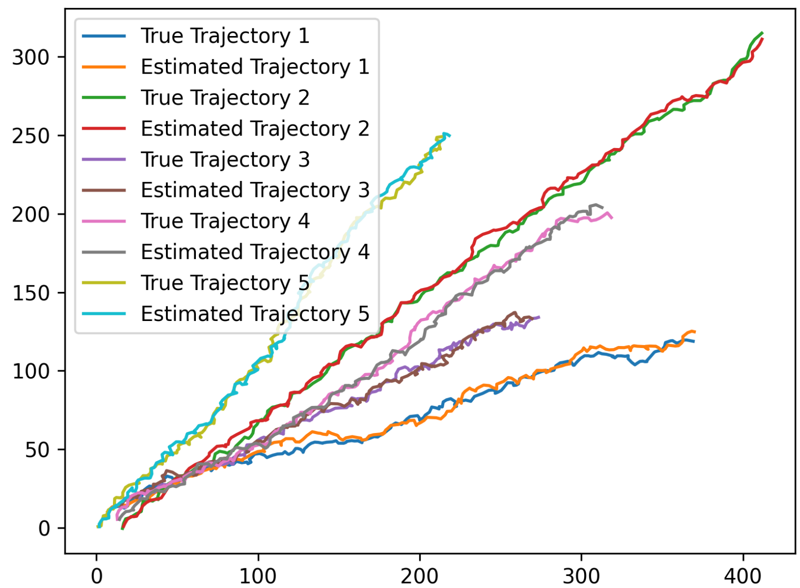

Figure 5: Trajectory Estimation with 5 agents and dataassociation

We ran many simulation for different starting points and initial velocities and observed that the algorithm is very robust and is able to predict the correct trajectory of each of the planes even when they are very close to each other.

2 Extended Kalman Filter

2.1 Model

Observe that the distance from a landmark is a non linear observation in terms of the state. Formally we can write,

$$U_t' = h_t(X_t) + \epsilon \text{ where } h_t(X_t) = ||X_t - landmark||$$

This can be incorporated using the **Extended Kalman Filter**. We linearize the above equation around the mean of the current estimated state.

$$\begin{aligned} h_t(X_t(= (x, y))) &= \sqrt{(x - l_0)^2 + (y - l_1)^2} \\ H &= \begin{bmatrix} \partial H_1/\partial x & \partial H_1/\partial y & 0 & 0 \end{bmatrix} \\ \partial H_1/\partial x = (x - l_0)/h_t, &\ \partial H_1/\partial y = (y - l_1)/h_t \end{aligned}$$

The third and fourth entries in H are zero since the distance will be partially independent of the velocity.

2.1.1 Observation Step For EKF

$$\begin{aligned} K_t &= \Sigma_t^{(P)} H^T \left(H \Sigma_t^{(P)} H^T + Q_t \right)^{-1} \\ \mathbf{X}_t &= X_t^{(P)} - K_t h_t(X_t^{(P)}) + K_t z_t \\ \Sigma_t &= (I - K_t H) \Sigma_t^{(P)} \end{aligned}$$

We implemented **Extended Kalman Filter** by adding another observation step which takes the landmark and distance as an input and updates the current belief.

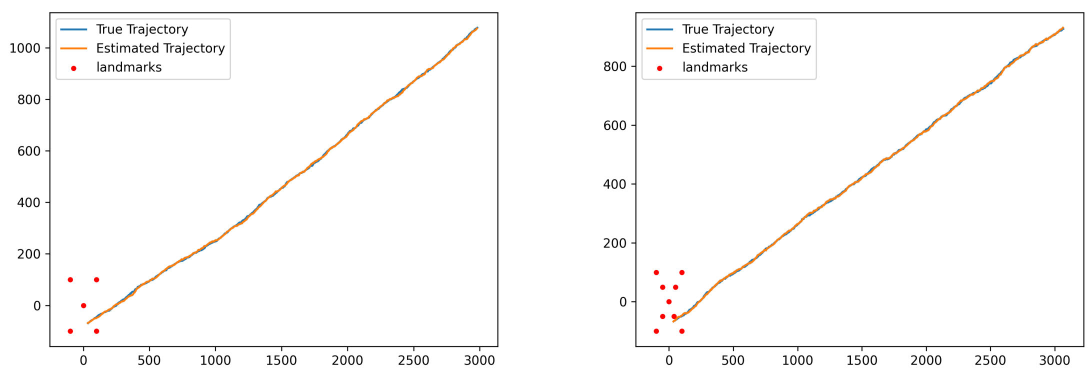

Figure 6: True and estimated trajectories with 0 action input for 5 (left) and 9 (right) landmarks

2.2 Uncertainty Estimates in position when the landmark is nearby

We observe that the uncertainty ellipse starts to shrinks when the plane comes near a landmark. This shows that adding more observations will decrease the uncertainty more as it will near to a landmark for more time.

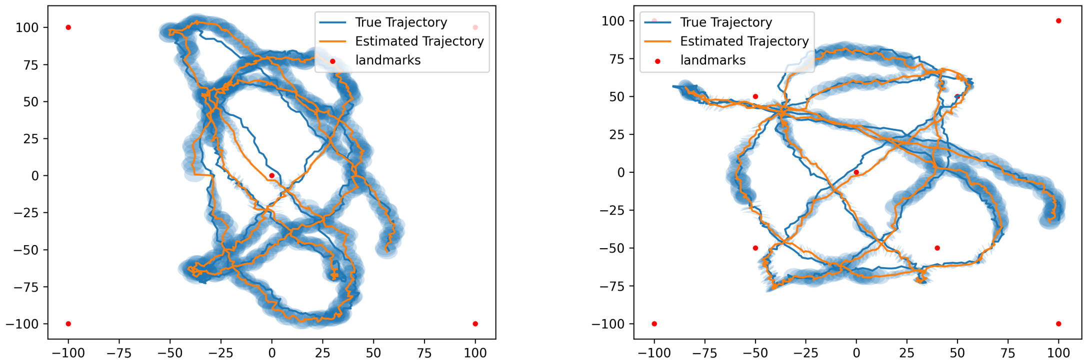

Figure 7: True and estimated trajectories with custom action input for 5 (left) and 9 (right) landmarks

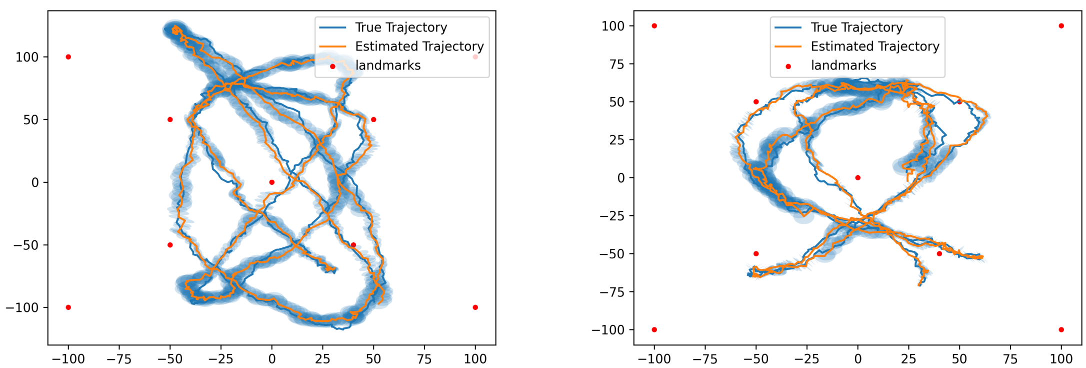

Figure 8: True and estimated trajectories with custom action input for more (left) and less (right) uncertainty in the landmarks' measurement

2.3 Effect of increasing uncertainty in the landmark measurements

We observe that as we increase the uncertainty in the landmark measurement (for example, making the standard deviation for landmark-distance observations equal to 10) , the uncertainty ellipse does not decrease much even after coming close to a landmark.

But if the uncertainty in the landmark measurement is decreased (for example, making the standard deviation for landmark-distance observations equal to 0.1), the uncertainty ellipse almost vanishes when the plane comes close to one of the landmarks.

2.4 Effects of adding additional landmarks

We observe that as we increase the number of landmarks, the average uncertainty gets decreased significantly over the whole path and the estimated trajectory becomes more similar to the actual trajectory which in turn helps in better tracking.

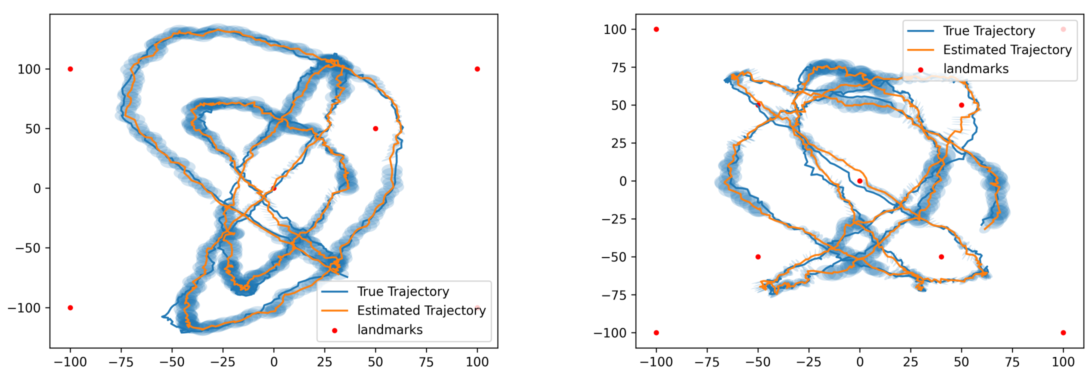

Figure 9: True and estimated trajectories with custom action input for less (left) and more (right) number of landmarks