Extended Kalman Filter Assignment 16-833

Harsh Sharma

October 23, 2020

Contents

1 Theory Questions

1.1 Next pose

$$P_{t+1} = g(P_t, u_t) \tag{1}$$

Where P_{t+1} is the predicted state at time t+1 and P_t is the state at time t. Assuming no noise/error in the control system and with control inputs d and .

$$\begin{gathered} x_{t+1} = x_1 + d_t \cos\theta \\ y_{1+1} = y_1 + d_t \sin\theta \\ \theta_{t+1} = \theta t + \alpha_t \\ \begin{pmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{pmatrix} = \begin{pmatrix} x_t \\ y_t \\ \theta_i \end{pmatrix} + \begin{pmatrix} d_t \cos\theta \\ d_t \sin\theta \\ \alpha_i \end{pmatrix} \end{gathered} \tag{2}$$

1.2 Uncertainty

Consider jacobian of $\partial g/\partial x$ as G-

$$G = \frac{\partial(x,u)}{\partial p_t} = \begin{pmatrix} \frac{\partial(g1)}{\partial x} & \frac{\partial(g1)}{\partial y} & \frac{\partial(g1)}{\partial \theta} \\ \frac{\partial(g2)}{\partial x} & \frac{\partial(g2)}{\partial y} & \frac{\partial(g2)}{\partial \theta} \\ \frac{\partial(g3)}{\partial x} & \frac{\partial(g3)}{\partial y} & \frac{\partial(g3)}{\partial \theta} \end{pmatrix} = \begin{pmatrix} 1 & 0 & -d_t \sin\theta \\ 0 & 1 & d_t \cos\theta \\ 0 & 0 & 1 \end{pmatrix}$$

If R is the process noise associated with each pose element as -

$$\begin{pmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\alpha^2 \end{pmatrix} \tag{3}$$

The co-variance associated with the pose can be given by-

$$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t \tag{4}$$

1.3 Estimated landmark position

Bearing angle is given by β which lies in the interval $[-\pi, \pi]$

Range - r. Estimated position of the landmark lx ly in global coordinates:

$$
\begin{aligned}
l_x &= x_t + (r + n_r)\cos(n_\beta + \beta + \theta_t) \\
l_y &= y_t + (r + n_r)\sin(n_\beta + \beta + \theta_t)
\end{aligned} \tag{5}
$$

where n_β and n_r are the zero mean Gaussian noise $\mathcal{N}\left(0, \sigma_\beta^2\right)$ and $\mathcal{N}\left(0, \sigma_r^2\right)$ associated with bearing and range measurement respectively.

1.4 Bearing Range measurement

If we know the position of the landmark (l_x, l_y) in global coordinates then the following equations are used to calculate the range and bearing.

$$
\begin{pmatrix} r \\ \beta \end{pmatrix} = \begin{pmatrix} \sqrt{(l_x - x)^2 + (l_y - y)^2} + n_r \\ \operatorname{atan} 2((l_y - y), (l_x - x)) - \theta_t - n_\beta \end{pmatrix}
$$

The bearing angle is wrapped between between $(-\pi, \pi)$. using warpToPI() function.

1.5 $\mathbf{H}_p$

Finding the measurement Jacobian H_p

$$
\delta = \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} = \begin{pmatrix} l_{x,j} - x_t \\ l_{y,j} - y_t \end{pmatrix} \tag{6}
$$

$$
\begin{gathered}
q = (l_x - x)^2 + (l_y - y)^2 \\
H_p = \frac{\partial(r,\beta)}{\partial p_t} = \begin{pmatrix} \frac{\partial(r)}{\partial x} & \frac{\partial(r)}{\partial y} & \frac{\partial(r)}{\partial \theta} \\ \frac{\partial(\beta)}{\partial x} & \frac{\partial(\beta)}{\partial y} & \frac{\partial(\beta)}{\partial \theta} \end{pmatrix} = \frac{1}{q} \begin{pmatrix} -\sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 \\ \delta_y & -\delta_x & -q \end{pmatrix}
\end{gathered} \tag{7}
$$

1.6 $\mathbf{H}_l$

Finding the measurement Jacobian with respect to the landmark

$$
H_l = \frac{\partial(r,\beta)}{\partial l} = \begin{pmatrix} \frac{\partial(r)}{\partial l_y} & \frac{\partial(r)}{\partial l_y} \\ \frac{\partial(\beta)}{\partial l_x} & \frac{\partial(\beta)}{\partial l_y} \end{pmatrix} = \frac{1}{q} \begin{pmatrix} \sqrt{q}\delta_x & \sqrt{q}\delta_y \\ -\delta_y & \delta_x \end{pmatrix}
$$

Parameters	Values
σ_x	0.25
σ_y	0.1
σ_α	0.1
σ_r	0.16
σ_β	0.1

We assume that all the landmark poses are independent of each other and that changing one would not affect the other.
For this reason, we calculate measurement jacobian with respect to the current landmark and not any other landmarks.

2 EKF implementation

2.1 Total landmarks

Total 6 landmarks are observed in the entire sequence as we observe 12 measurements of range,bearing for all of them.

2.2 Final Trajectory

For the parameters shown in the table below , the final trajectory obtained can be seen in the figure 1.

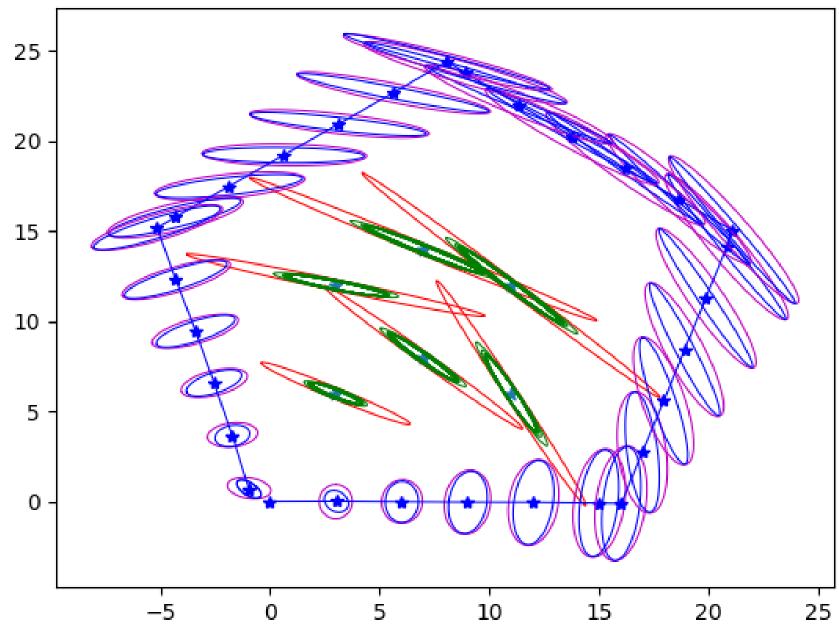

Figure 1: Final Trajectory

2.3 EKF Explained

The robot moves and the prediction is made for its pose using the predict step. This increases the uncertainty of the robot's position (magenta eclipse).

In the next step measurement is taken for the landmarks and update step is used. This decreases the pose uncertainty (blue eclipse).
Every time new measurements are added to the system new information is being added and that results to more and more accurate estimations. Similarly for the landmarks, the initial uncertainty corresponding to their the landmarks (red eclipse) decreases with more data. (green eclipse)
This can be seen in the figure 7 below.
All of the above assumes reasonable initializations of state uncertainties and means i.e. landmark,robot poses and their covariances.

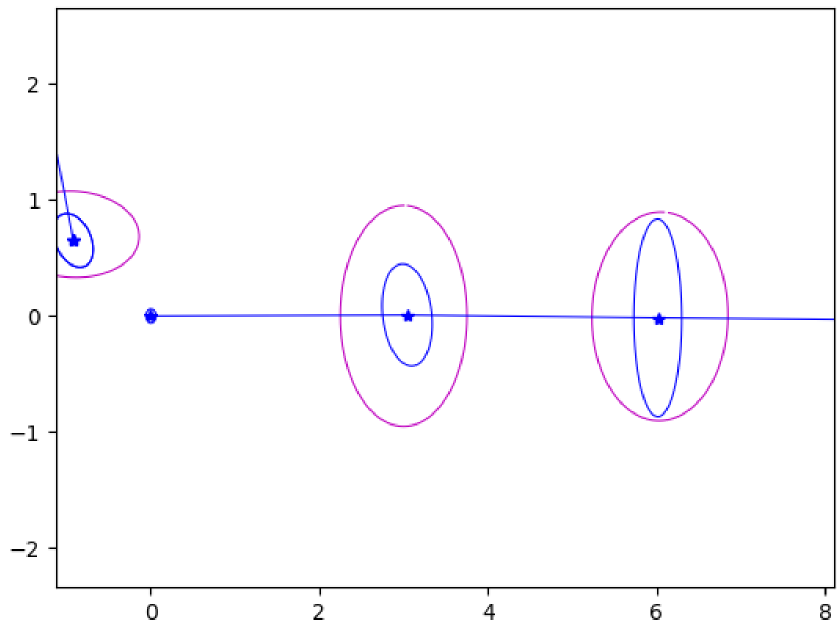

Figure 2: After Iteration 1

2.3.1 Initialization

We initialize the covariance of our state vector **X** which is a concatenation of robot and landmark poses(in this case happens to be 15x1 vector).

We initialize covariance as having diagonal elements of covariance matrix of robot poses as 0. Tha landmark means and covariances are found from initial measurement reading of range and bearings comprising to each landmark and initial robot pose covariance.

State Covariance-

$$\Sigma_0 = \ P \left[\begin{array}{c|c} \Sigma_{x_0} & 0 \\ 0 & \Sigma_{l_0} \end{array} \right] \tag{8}$$

$$\Sigma_{l_0}$$

here is a 12x12 block diagonal matrix found from initial readings of landmark and bearings.

$$\Sigma_{x_0} = diag[\ \sigma_x^2 \quad \sigma_y^2 \quad \sigma_\theta^2\ ] \tag{9}$$

2.3.2 EKF Prediction

$$\bar{X}_t = g(X_{t-1}, u_t) \tag{10}$$

$$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t \tag{11}$$

2.3.3 EKF Measurement

There are multiple update steps corresponding to measurement from each landmark. Every time Kalman gain K is computed and state means and covariances are updated-

$$\begin{aligned} \bar{x}_t &= \bar{x}_t + K\left(z - z_t\right) \\ \bar{\Sigma}_t &= (I - K * H)\bar{\Sigma}_t \end{aligned} \tag{12}$$

2.4 Evaluation

Estimated positions of the landmarks from ground truth are in the table below-

Euclidean	**Mahalanobis**
0.00193	0.0246
0.0032	0.0152
0.0024	0.0416
0.0035	0.0451
0.0029	0.032
0.00401	0.0676

Yes each of them is inside the smallest eclipse as shown in figure 3 which means that the estimated landmark poses are correct.

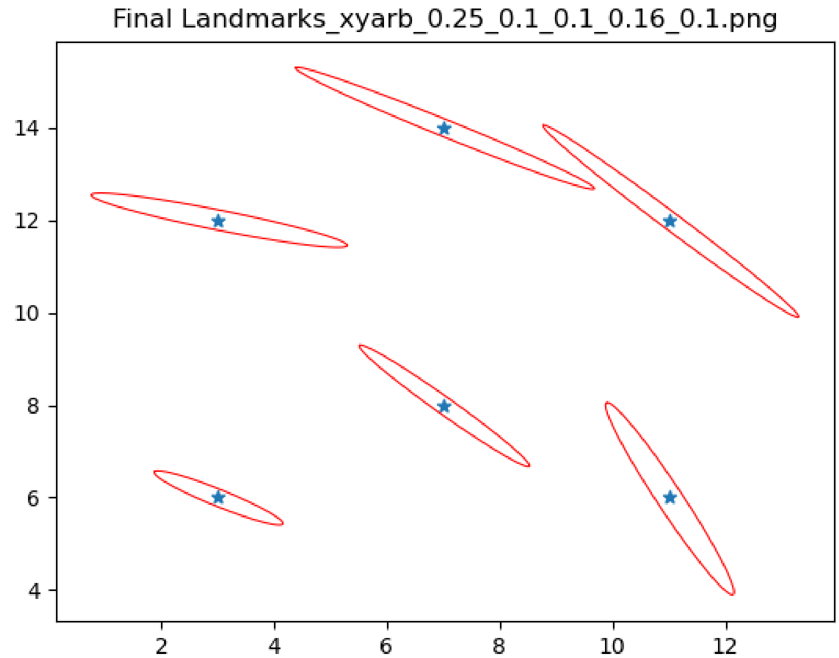

Figure 3: Final groundtruth (*) landmarks plotted with the final covariance(red ellipses) associated with each of them after EKF iterations

3 Discussion

3.1 Covariance matrix sparsity

1. We assume cross covariances to be zero for the poses, which is not certainly true. Uncertainty in x direction can affect the robot's uncertainty in y as well as its yaw.
2. Similarly we assume that the landmarks are independent of each other.
3. We also assume robot poses and landmarks are not correlated.

However, during the update steps the K_t becomes non sparse and full. This alters the non-diagonal entries of the state's covariance matrix.
So this suggests that the landmarks are correlated, or poses and landmarks are correlated. Assuming that the landmarks are independent was the wrong assumption in the beginning. Knowing about one landmark improves the estimation of other landmarks.

3.2 Parameters change

3.2.1 Experiment 1

Increase in sigma of poses-

This increases the covariance associated with the robot poses as shown in figure below.

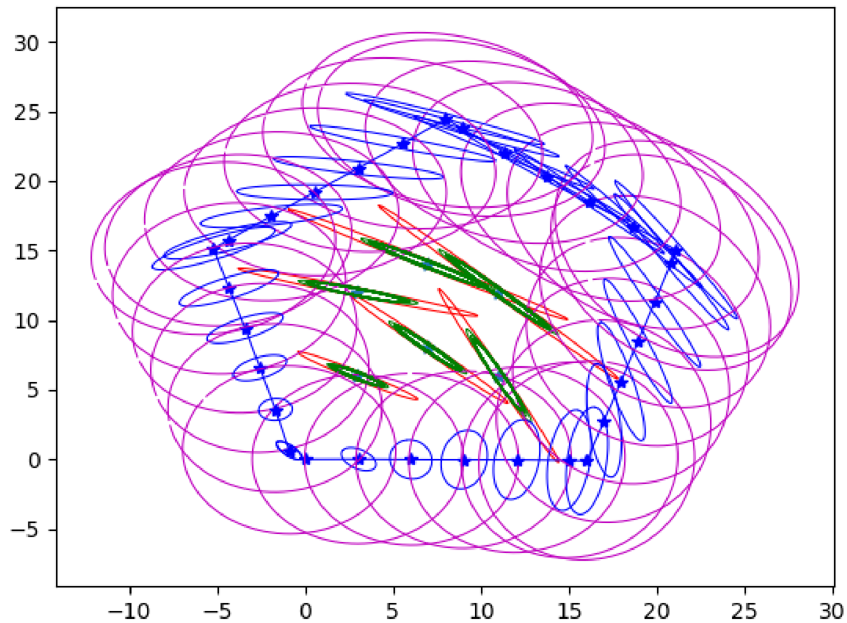

Figure 4: Increasing pose sigmas

3.2.2 Experiment 2

Decrease in sigma of poses-

This decreases the covariance associated with the robot poses as shown in figure below.

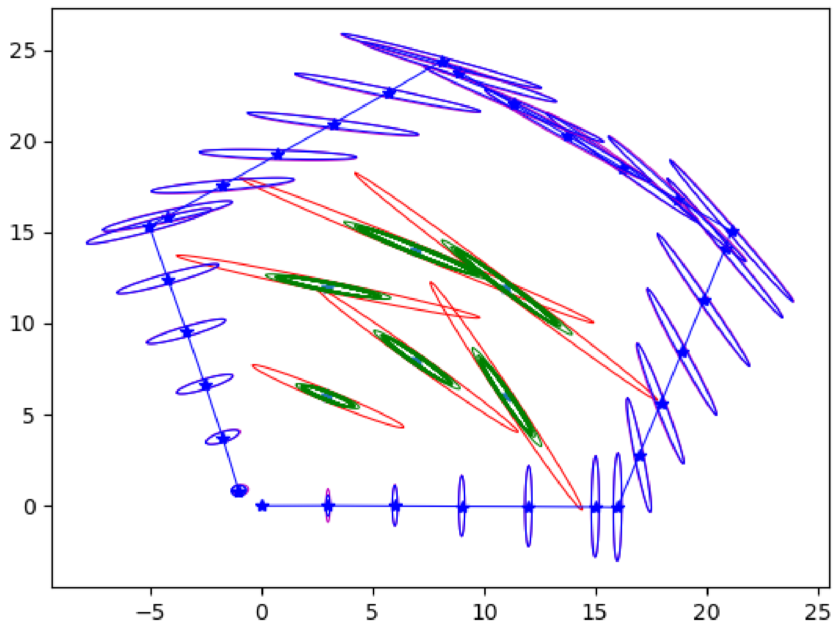

Figure 5: Decreasing pose sigmas

3.2.3 Experiment 3

Increasing the sensor variances σ_β and σ_r increases the noise corresponding to the landmarks.

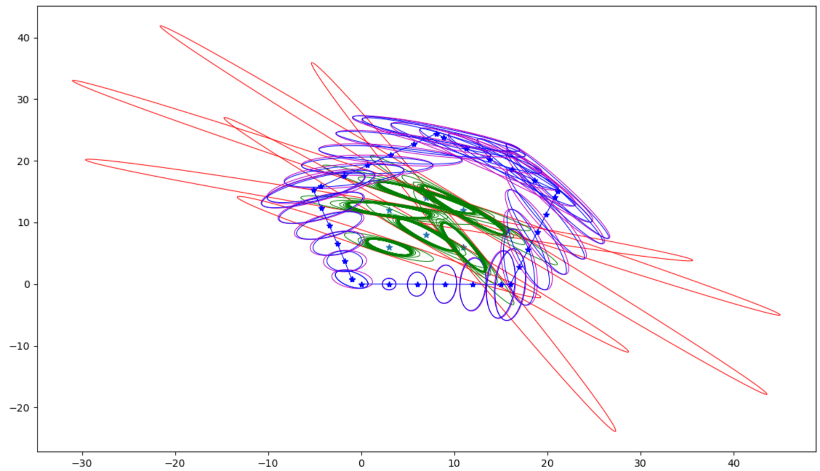

Figure 6: Increasing sensor variances

3.2.4 Experiment 4

Decreasing the sensor variances too much causes it to be overconfident of its landmarks which is wrong which deviates the robot from its trajectory and constructed trajectory is completely wrong. This is why it is important to chooses the initialisations correctly.

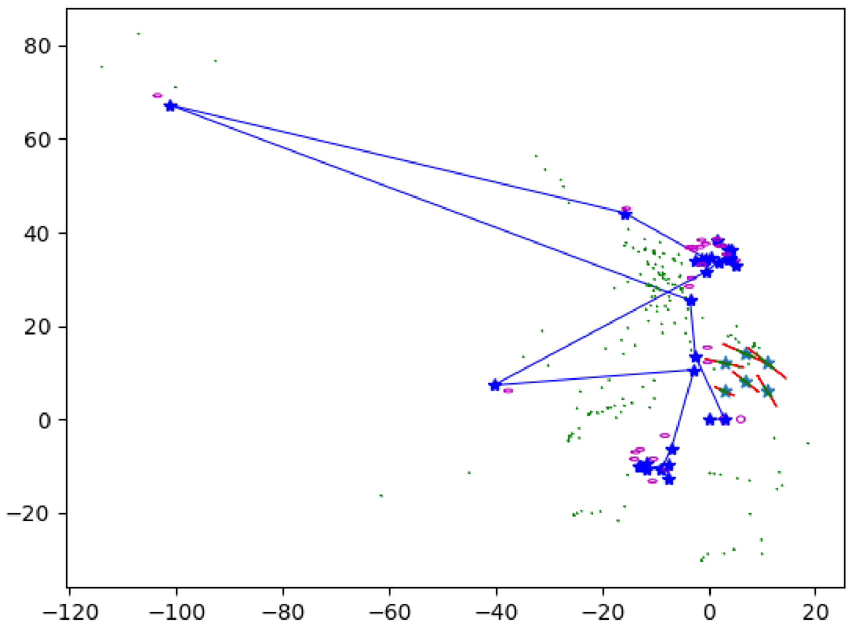

Figure 7: Decreasing sensor variances too much

3.3 Speed up EKF

The following methods can speed up the EKF process-

- Fixing the number of landmarks though updating key landmarks. This will keep the time same but heuristic to chose key landmarks(or new landmarks based on new position) can help us better estimate.
- Maintaining updates for local and global map seperately. Every time we can decide on solving the problem independently for a local submap. Via heuristics we can know when to switch to a different submap. Later we ca nhave optimisation for stitching the local submaps together done offline.
- Random Sampling of landmarks to keep the total number of landmarks bounded for each update
- Optimal factorization of the matrix $(H\bar{\Sigma}H^T + R)$ which can reduce the complexity of EKF which now requires to invert a matrix in $O(n^3)$.

- Additional heuristics like identifying moving objects and removing them from the landmarks.