Carnegie Mellon University

Spring 2022

16-833: Simultaneous Localization and Mapping

HW2 Extended Kalman Filter

Prof. Michael Kaess

Chinmay Garg

`Andrew ID : chinmayg`

2022-03-15

Theory

A robot is moving on the 2D ground plane. In each time step t, the robot is controlled to move forward (the x-direction of the robot's coordinates) d_t meters, and then rotate θ_t radian. The pose of the robot in the global coordinates at time t is written as a vector $p_t = \begin{bmatrix} x_t & y_t & \theta_t \end{bmatrix}^T$, where x_t and y_t are the 2D coordinates of the robot's position, and θ_t is the robot's orientation.

1. Predicted State $t+1$

With the given p_t, the predicted state p_{t+1} can be calculated using the non-linear function $g(x,u)$ as :

$$p_{t+1} = g(x,u) \tag{1}$$

$$p_{t+1} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + \begin{bmatrix} d_t \cos\theta_t \\ d_t \sin\theta_t \\ \alpha_t \end{bmatrix} = \begin{bmatrix} x_t + d_t \cos\theta_t \\ y_t + d_t \sin\theta_t \\ \theta_t + \alpha_t \end{bmatrix} \tag{2}$$

2. Predicted Uncertainty

The Jacobian of the non-linear function $g(x,u)$, G_t:

$$G_t = \nabla_g J = \begin{bmatrix} 1 & 0 & -d_t \sin\theta \\ 0 & 1 & d_t \cos\theta \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

With the errors following Gaussian Distributions, $e_x \sim \mathcal{N}(0, \sigma_x^2)$, $e_y \sim \mathcal{N}(0, \sigma_y^2)$, and $e_\theta \sim \mathcal{N}(0, \sigma_\theta^2)$, process noise R_t:

$$R_t = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\alpha^2 \end{bmatrix}$$

With the process noise and the uncertainty at time t given by $\mathcal{N}(0, \Sigma_t)$, the predicted uncertainty for the motion model at time $t+1$ can be given by equation 4 and the expected state remains the same as noiseless state represented in equations 1 and 2.

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

3. Landmarks

The location of the landmark l_x and l_y, for a range measurement r and bearing measurement β with the zero mean Gaussian Noises $\eta_r \sim \mathcal{N}(0, \sigma_r^2)$ and $\eta_\beta \sim \mathcal{N}(0, \sigma_\beta^2)$ are:

$$l_x = x_t + (r + \eta_r)\cos(\theta_t + \beta + \eta_\beta)$$
$$l_x = y_t + (r + \eta_r)\sin(\theta_t + \beta + \eta_\beta)$$

4. Predicted Landmarks

With the landmark position (l_x, l_y) known in the global coordinate frame, the predicted measurement r_{pred} and β_{pred} can be calculated as (here, robot state is (x, y, θ):

$$r_{pred} = \sqrt{(l_x - x)^2 + (l_y - y)^2} + \eta_r$$

$$\beta_{pred} = \arctan 2(\frac{l_y - y}{l_x - x}) - \theta + \eta_\beta$$

$$\beta_{pred} = wrap2pi(\beta_{pred})$$

where the $wrap2pi$ function wraps the value of β within $(-\pi, \pi]$.

5. Jacobian wrt. Robot Pose

Jacobian $H_p = \frac{\partial Z_{pred}}{\partial p_t} = \begin{bmatrix} \frac{\partial r_{pred}}{\partial x} & \frac{\partial r_{pred}}{\partial y} & \frac{\partial r_{pred}}{\partial \theta} \\ \frac{\partial \beta_{pred}}{\partial x} & \frac{\partial \beta_{pred}}{\partial y} & \frac{\partial \beta_{pred}}{\partial \theta} \end{bmatrix} = \begin{bmatrix} -\frac{l_x - x}{D} & -\frac{l_y - y}{D} & 0 \\ \frac{l_y - y}{D^2} & -\frac{l_x - x}{D^2} & -1 \end{bmatrix}$

where $D = \sqrt{(l_x - x)^2 + (l_y - y)^2}$

6. Jacobian wrt. Landmark

Jacobian $H_l = \frac{\partial Z_{pred}}{\partial l} = \begin{bmatrix} \frac{\partial r_{pred}}{\partial l_x} & \frac{\partial r_{pred}}{\partial l_y} \\ \frac{\partial \beta_{pred}}{\partial l_x} & \frac{\partial \beta_{pred}}{\partial l_y} \end{bmatrix} = \begin{bmatrix} \frac{l_x - x}{D} & \frac{l_y - y}{D} \\ -\frac{l_y - y}{D^2} & \frac{l_x - x}{D^2} \end{bmatrix}$

where $D = \sqrt{(l_x - x)^2 + (l_y - y)^2}$

Further, we do not calculate the measurement Jacobian with respect to other landmarks because we assume that the landmarks are independent of each other, thus they don't influence each other. So, even if we calculate a Jacobian for other landmarks we will get 0 values in the Jacobian, whereas we get non-zero values for the current landmark.

EKF Implementation

Problem 1

There are 6 landmarks in the environment which are being observed over the entire sequence.

Problem 2

On implementing the EKF Algorithm for the given data, the visualized path and the landmarks can be seen in figure 1 and figure 2. The trajectory in figure 1 has been generated using the default test parameters in table 7.

σ_x	0.25
σ_y	0.1
σ_α	0.1
σ_β	0.01
σ_r	0.08

Table 1: Default Parameters

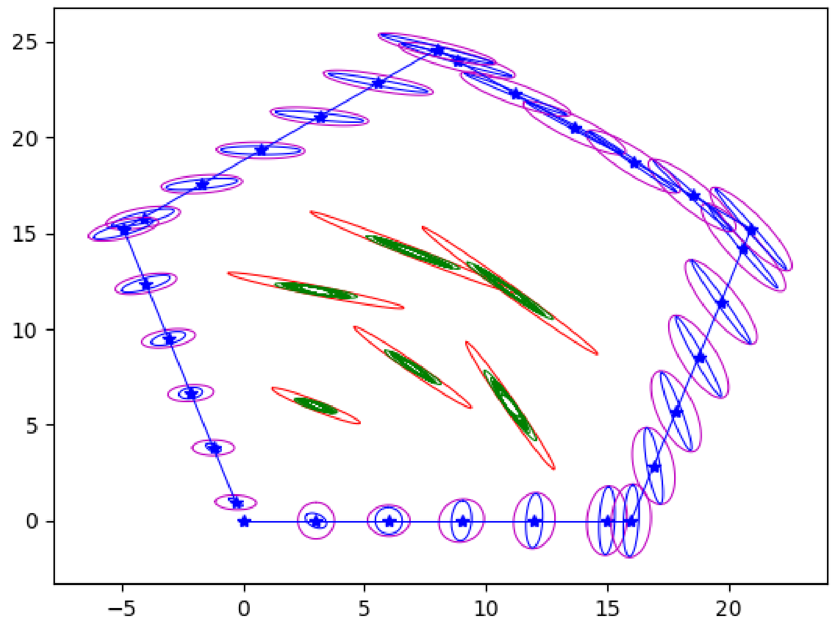


Figure 1: Trajectory of the robot and landmarks in the middle

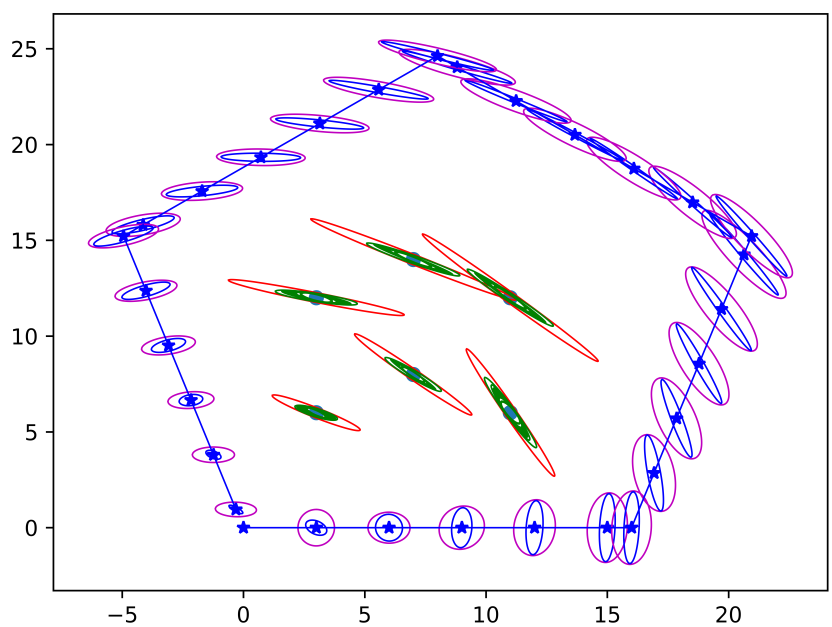


Figure 2: Plot with true landmark positions from evaluation

Problem 3

After the first measurement is used to initialize the mean and covariances of the landmarks (shown in red), the successive measurements update the position of the landmarks as the robot moves around. Initially, we see that the uncertainty with the farthest landmark is the highest and the closest landmark is the lowest as in figure 3.

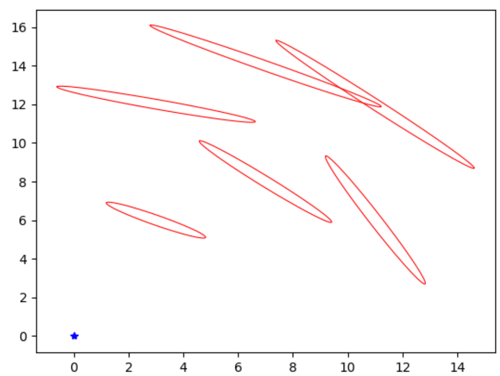


Figure 3: Initial Landmark Uncertainty Distribution of Landmarks

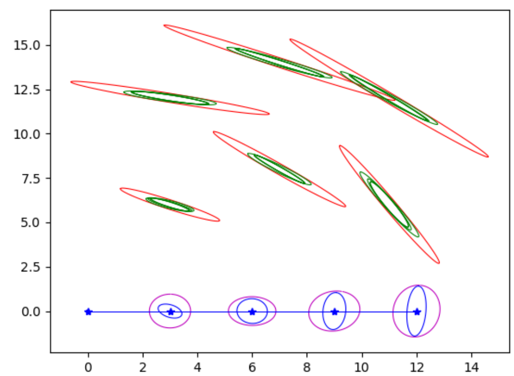


Figure 4: Predicted and Updated uncertainties of Landmarks and Robot

As the robot moves on it's trajectory, for each step, first the uncertainty for it's position is predicted (shown in magenta which includes its expected position and covariance. This uncertainty at the new step is larger than before due to added uncertainty with movement. Then to update the uncertainty of the robot with the new information (odometry and landmark measurements), we iterate over the different landmarks to use their measurements to calculate the Kalman gain of the covariance matrix and update the robot's pose and uncertainty (here the pose include the robot's position and the landmarks position).

For each landmark we predict a range and bearing using the actual measurement and the robot's expected position. The predicted measurement and the actual measurement are then used to find the error, which is used to correct the position of robot and landmarks, along with their covariances using the Kalman gain in the update step iteratively.

Once we update the expected pose and covariance in the update step, we see that the distribution becomes smaller for both the robot and landmarks (updates shown in green for the landmarks and blue for robot) as shown in figure 4 which is what is expected. Further as the robot moves on it's trajectory the uncertainty keeps changing, and reducing with the final uncertainty being quite low for both landmarks and robot.

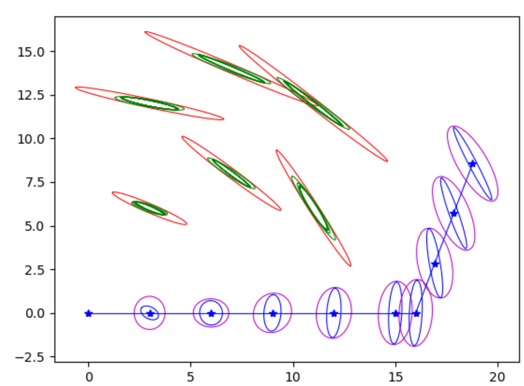


Figure 5: Rotational uncertainty in distributions

Problem 4

The computed distances can be seen in the table 2. The distances indicate how far is the current expected position (prediction) from the actual ground truth position of the landmarks in the global frame.

Landmark	**Mahalanobis Distance**	**Euclidean Distance**
1	0.0037	0.0022
2	0.0569	0.0119
3	0.0075	0.0031
4	0.0009	0.0014
5	0.0481	0.0123
6	0.0057	0.0029

Table 2: Mahalanobis Distance and Euclidean Distance of landmarks from expected positions

Discussion

Problem 1

The zero terms in the initial landmark covariance matrix are the cross-covariances between different landmarks. Over the the limit, all the landmarks become correlated therefore in the final state covariance matrix the cross-covariances are non-zero. This happens because as the robot traverses around the landmarks in the environment, it adds correlations between its pose and the landmarks that it sees to improve the pose estimate. In the limit, observing a landmark improves the robot pose estimate and consequently it also eliminates some of the uncertainty of landmarks previously seen by the robot. The information that helps localize the robot is propagated through the map. Thus, essentially adding correlations between the landmarks, which leads to non-zero cross-covariances.

When setting the initial covariances of the landmarks, an assumption was made that the landmarks are independent of each other and don't influence each other's position. Thus, initially, only the self-covariances i.e. variances were modeled for the distribution of the landmarks and therefore the cross-covariances are initialized to 0 based on this assumption.

Problem 2

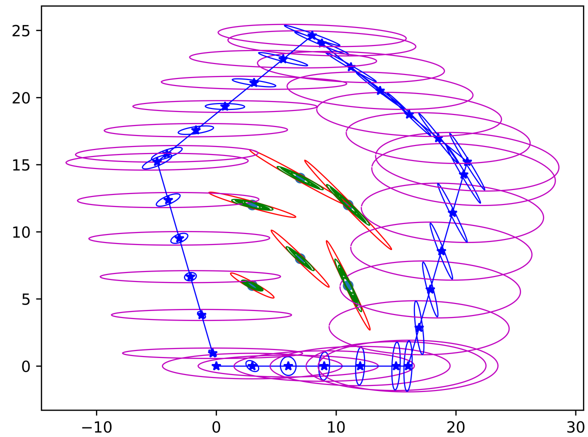
The following are the results of modifications of different uncertainty parameters.

Parameter σ_x

Increasing the uncertainty in x by 10 times, leads to higher variance along the x-axis in robot's pose which persists to the end of the trajectory.

σ_x	2.5
σ_y	0.1
σ_α	0.1
σ_β	0.01
σ_r	0.08

Table 3: Parameters

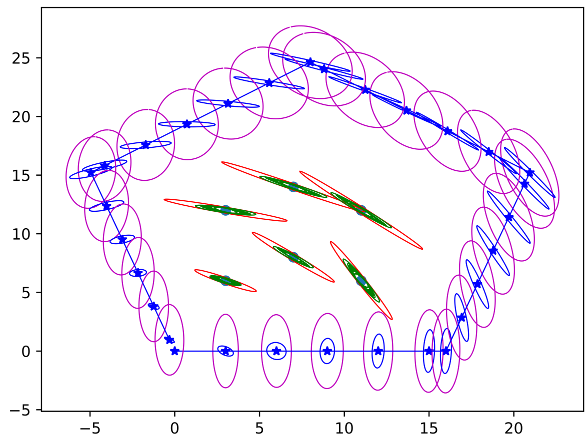


Parameter σ_y

Increasing the uncertainty in y by 10 times, leads to higher variance along the y-axis in robot's pose which persists to the end of the trajectory.

σ_x	0.25
σ_y	1
σ_α	0.1
σ_β	0.01
σ_r	0.08

Table 4: Parameters

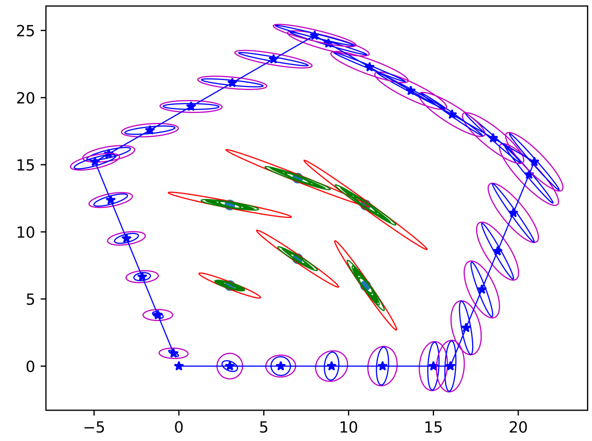


Parameter σ_α

Increasing the uncertainty in α by 10 times, doesn't have a huge effect on the uncertainty of the robot's pose except for a marginally greater uncertainty in the pose.

σ_x	0.25
σ_y	0.1
σ_α	1
σ_β	0.01
σ_r	0.08

Table 5: Parameters

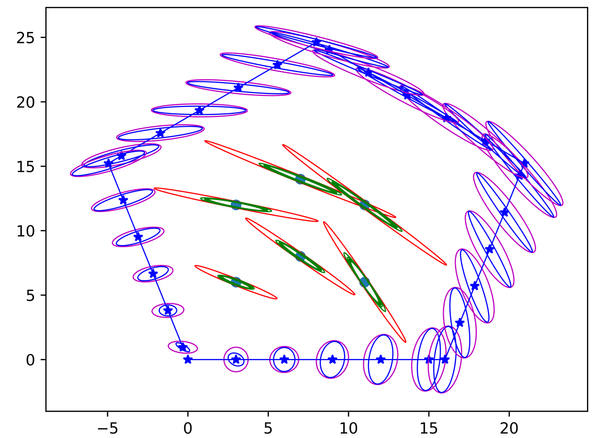


Parameter σ_β

Increasing the noise variance in the bearing measurement increases the uncertainty distribution of the landmarks perpendicular to the direction towards the robot.

σ_x	0.25
σ_y	0.1
σ_α	0.1
σ_β	0.01
σ_r	0.8

Table 6: Parameters

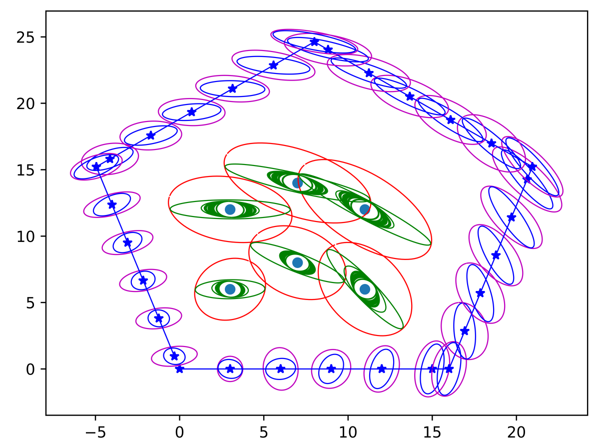


Parameter σ_r

Increasing the noise variance in the range measurement increases the uncertainty distribution of the landmarks along the direction towards the robot.

σ_x	0.25
σ_y	0.1
σ_α	0.1
σ_β	0.1
σ_r	0.08

Table 7: Parameters



Problem 3

To ensure EKF SLAM system achieve constant time or overcome the issue of increased landmarks by faster computation, the following solutions can be implemented:

- The robot can consider a fixed number of landmarks when localizing itself and studying the environment. The latest 'n' landmarks can be used to compute the expected pose.
- The state can be divided into sub-processes. Thus only the state variables that occur in the measurements at each time step are updated, and the other variables that are not a part of the measurements are not updated. Therefore, only a fixed number of variables are updated at each step and the whole state is only updated occasionally.
- Optimizing the inversion of matrix during the update step, can reduce the time complexity of the operation making EKF SLAM faster.
- Among the landmarks, it is possible that some landmarks are better than the others for localization so we create an isolated map containing only these landmarks (primary). With the positions of these, then occasionally we update all the other landmarks (secondary) in the map relative to the landmarks (primary) in the isolated map.