16-833: Robot Localization and Mapping, Spring 2021

Homework 2-SLAM using Extended Kalman Filter

Aaron Guan (zhongg)

1 Theory

1. The next pose $\mathbf{p}_{t+1}$ is:

$$\mathbf{p}_{t+1} = \mathbf{p}_t + \begin{bmatrix} d_t \cos(\theta_t) \\ d_t \sin(\theta_t) \\ \alpha_t \end{bmatrix} = \begin{bmatrix} x_t + d_t \cos(\theta_t) \\ y_t + d_t \sin(\theta_t) \\ \theta_t + \alpha_t \end{bmatrix}$$

2. Using Kalman state prediction equation, we have:

$$\Sigma_{t+1} = G_t \Sigma_t G_t^T + R_t$$

The Jacobian of the pose with respect to the state at time t is given by:

$$G_t = \begin{bmatrix} 1 & 0 & -d_t \sin(\theta_t) \\ 0 & 1 & d_t \cos(\theta_t) \\ 0 & 0 & 1 \end{bmatrix}$$

To get R_t, we need to covert the noise parameters of the system in robot's coordinates R to global coordinate using H_t:

$$R = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\alpha^2 \end{bmatrix} H_t = \begin{bmatrix} \cos(\theta_t) & -\sin(\theta_t) & 0 \\ \sin(\theta_t) & \cos(\theta_t) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_t = H_t R H_t^T$$

the predicted uncertainty of the robot at time $t+1$ would be:

$$\mathcal{N}(0, \Sigma_{t+1}) = G_t \mathcal{N}(0, \Sigma_t) G_t^T + R_t$$

3. The global position of the landmarks is:

$$\mathbf{L} = \begin{bmatrix} l_x \\ l_y \end{bmatrix} = \begin{bmatrix} x_t + r\cos(\theta_t + \beta) \\ y_t + r\sin(\theta_t + \beta) \end{bmatrix}$$

Given the noise in the measurements $\eta_\beta \sim \mathcal{N}\left(0, \sigma_\beta^2\right)$ and $\eta_r \sim \mathcal{N}(0, \sigma_r^2)$, the estimates of the landmark would be:

$$\mathbf{L} = \begin{bmatrix} l_x \\ l_y \end{bmatrix} = \begin{bmatrix} x_t + (r + \eta_r)\cos(\theta_t + \beta + \eta_\beta) \\ y_t + (r + \eta_r)\sin(\theta_t + \beta + \eta_\beta) \end{bmatrix}$$

4. Given that we know the landmark l is at (l_x, l_y) and the robot state at time t is $\mathbf{p}_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}$, we can define the distance between robot and landmark in global frame is $\delta_x = l_x - x_t$ and $\delta_y = l_y - y_t$. We can define the quantity $q = \delta_x^2 + \delta_y^2$. Then we can get the predicted measurement of bearing and range:

$$h(p) = \begin{bmatrix} \beta \\ r \end{bmatrix} = \begin{bmatrix} warp2pi\left(atan2\left(\delta_y, \delta_x\right) - \theta_t\right) \\ \sqrt{q} \end{bmatrix}$$

5. Computing the Jacobian H_p with respect to the state (x, y, θ):

$$H_p = \begin{bmatrix} \frac{\partial h_1}{\partial x_t} & \frac{\partial h_1}{\partial y_t} & \frac{\partial h_1}{\partial \theta_t} \\ \frac{\partial h_2}{\partial x_t} & \frac{\partial h_2}{\partial y_t} & \frac{\partial h_2}{\partial \theta_t} \end{bmatrix} = \frac{1}{q}\begin{bmatrix} \delta_y & -\delta_x & -q \\ -\sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 \end{bmatrix}$$

where $\delta_x = l_x - x_t$, $\delta_y = l_y - y_t$, $q = \delta_x^2 + \delta_y^2$.

6. Computing the Jacobian H_l with resect to the landmarks:

$$H_l = \begin{bmatrix} \frac{\partial h_1}{\partial l_x} & \frac{\partial h_1}{\partial l_y} \\ \frac{\partial h_2}{\partial l_x} & \frac{\partial h_2}{\partial l_y} \end{bmatrix} = \frac{1}{q}\begin{bmatrix} -\delta_y & \delta_x \\ \sqrt{q}\delta_x & \sqrt{q}\delta_y \end{bmatrix}$$

where $\delta_x = l_x - x_t$, $\delta_y = l_y - y_t$, $q = \delta_x^2 + \delta_y^2$.

The reason why we do not need to calculate the measurement Jacobian with respect to other landmarks except for itself is based on the assumption that the measurements of landmarks with respect to the robot are independent to each other.

2 Implementation

1. The fixed number of landmarks being observed over the entire sequence is 6.

2. The final visualization showing the landmarks and robot trajectory is shown in Figure 1.

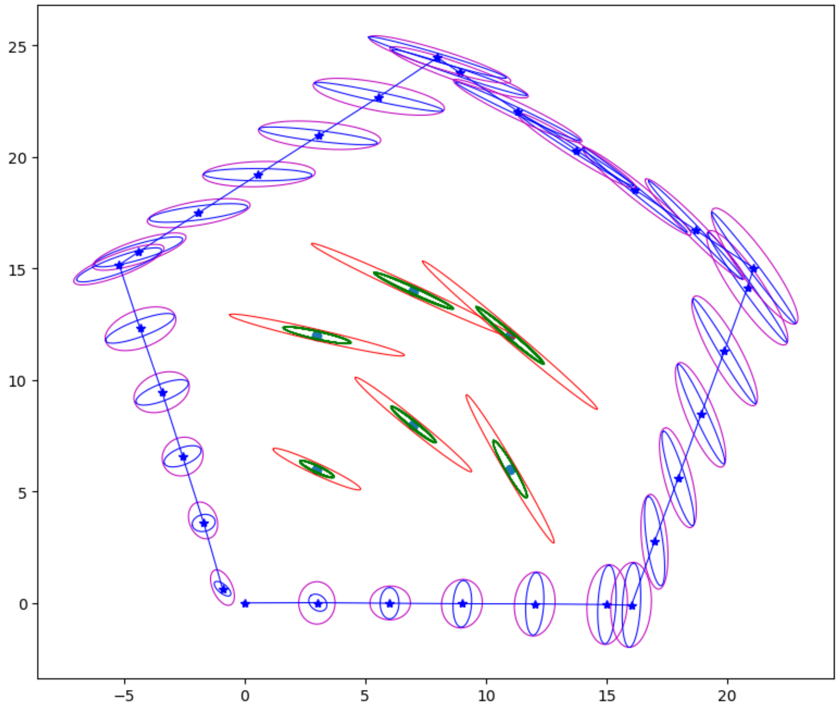

Figure 1: Visualization result

3. The EKF-SLAM algorithm improves the estimation of both the trajectory and the map by using the predicted covariance to compute the the Kalman gain K, which is a measurement that gives relative weight to the measurements and the current state estimate. The EKF-SLAM uses this Kalman gain to update the predicted belief and the covariance and the value of the Kalman gain depends on the uncertainty. If the Kalman gain is low, it means the system rely heavily on the accurate state estimation. When the uncertainty is high, the gain is inherently higher and the system would rely more on the measurement data to improve the estimation.

4. The final visualization with ground truth landmark positions marked in red are shown in Figure 2. The Euclidean and the Mahalanobis distance of the 6 landmarks are listed in Table 1.

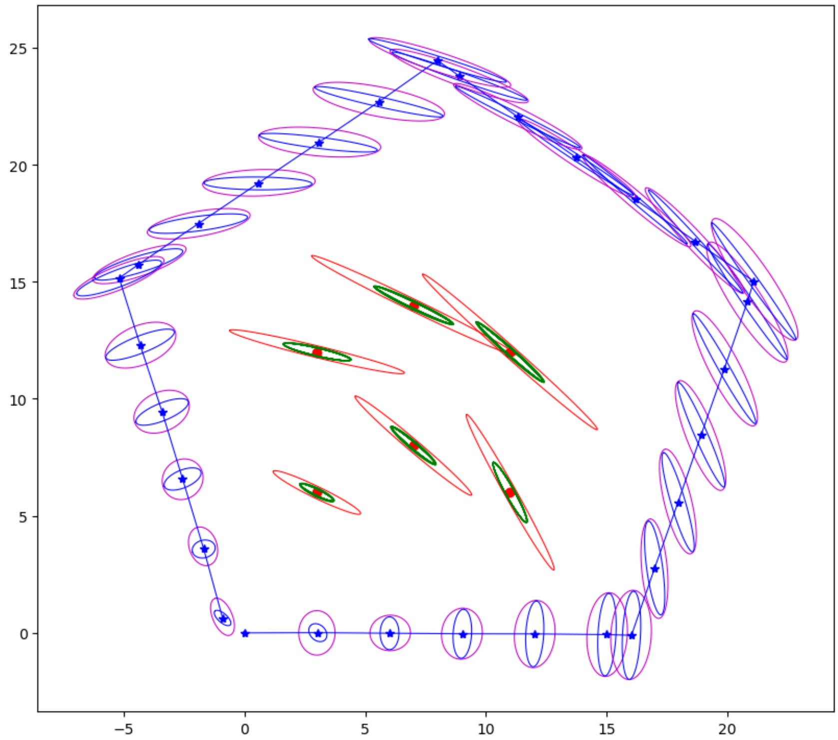

Figure 2: Visualization result with ground truth landmark position in red

Landmark #	Euclidean Distance	Mahalanobis Distance
1	0.002717634338694619	0.05242282796430613
2	0.003430806408451885	0.06216337123715574
3	0.004829431915684942	0.03731816820860023
4	0.005380537962684753	0.0643291110450752
5	0.004287175492425487	0.025078754058870622
6	0.004738530051409496	0.09536690959437051

Table 1: Euclidean and Mahalanobis distances for 6 landmarks

The ground truth point for all the landmarks are inside the smallest corresponding ellipse. It indicates that the updates of the EKF-SLAM are correct and the estimated positions of the landmarks has good distributions such that the ground truth can be sampled from the estimated distribution.

The Euclidean distance computes the absolute geometric distance in the spatial dimension. The small Euclidean distances indicates that the estimated position of landmarks are very close to the ground truth position in the absolute 2D space (x, y). The Mahalanobis distance measures the distance between a point P and a distribution D. It measures how many standard deviations away P is from the mean of D. The distance is zero if P is the mean of D. Since the numbers are small, it indicates that we get a good probability distribution of the landmarks.

3 Discussion

1. The reason why the zero terms in the initial landmark covariance matrix become non-zero in the final state covariance matrix is because they are updated during the update step of the EKF-SLAM. During the update step, the uncertainty will be incorporated into the estimation of the state and covariance, which creating uncertainties of each state elements relative to other elements.

   During the initialization of the state covariance matrix, we assume that each state element is independent to all other state elements such that we can form a diagonal covariance matrix.

2. Totally 5 additional experiments are performed and each parameter σ_x, σ_y, σ_α, σ_β, σ_r is set 10 times bigger at a time and others are fixed. As shown in Figure 3, when the σ_x is 10 times larger, the uncertainty of robot pose in x-axis becomes much bigger indicating the localization becomes more uncertain. The mapping remain almost the same since the σ_β and σ_r are not modified.

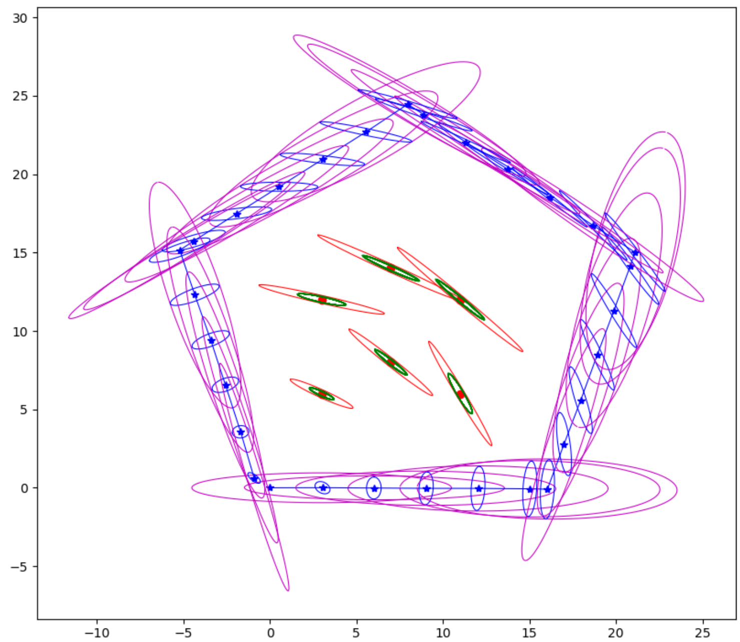

Figure 3: Visualization result with $10 \times \sigma_x$

   From the below Figure 4, the uncertainty of the robot pose in y-axis becomes larger when the σ_y is 10 times larger. Similar as changing the σ_x, modifying the σ_y will influence the localization capability and good observations become more important for robot to achieve good localization.

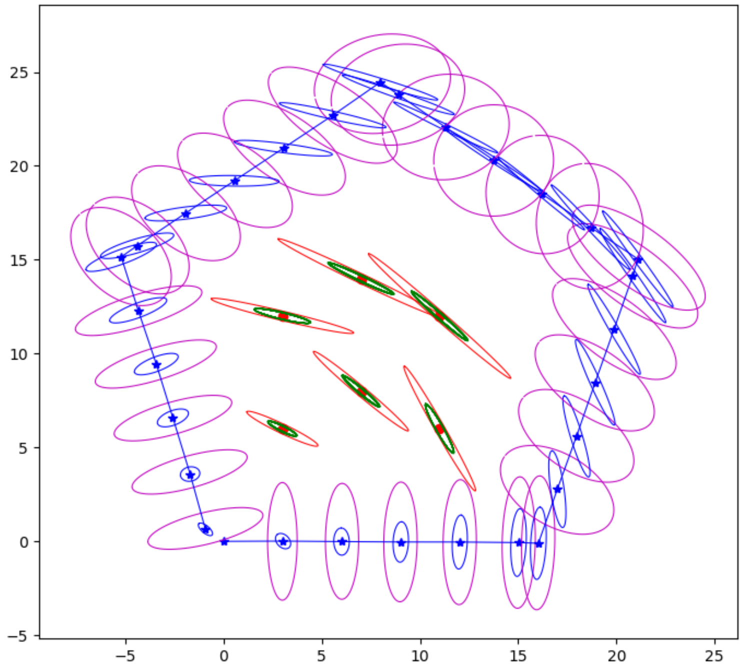

Figure 4: Visualization result with $10 \times \sigma_y$

The visualization result with the 10 times bigger σ_α is shown in Figure 5. The robot's trajectory and the landmark estimated position are almost the same as the result with the original set of parameters. In this problem case, it indicates that the uncertainty of the robot orientation is not big enough to affect the overall performance of the localization.

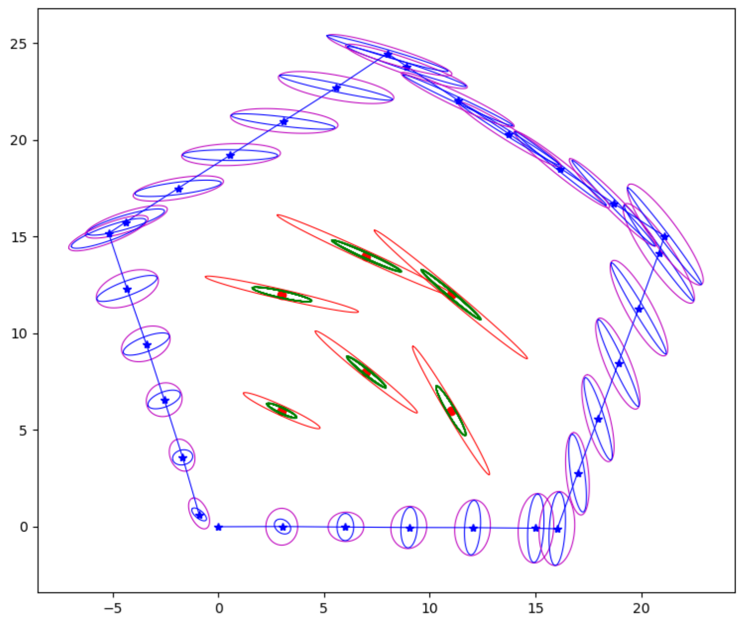

Figure 5: Visualization result with $10 \times \sigma_\alpha$

Figure 6 shows the result with the 10 times bigger σ_β. The uncertainty of the landmark becomes larger as the eclipse is much longer. Both predicted uncertainty and updated uncertainty of robot's position becomes slightly bigger than the original one.

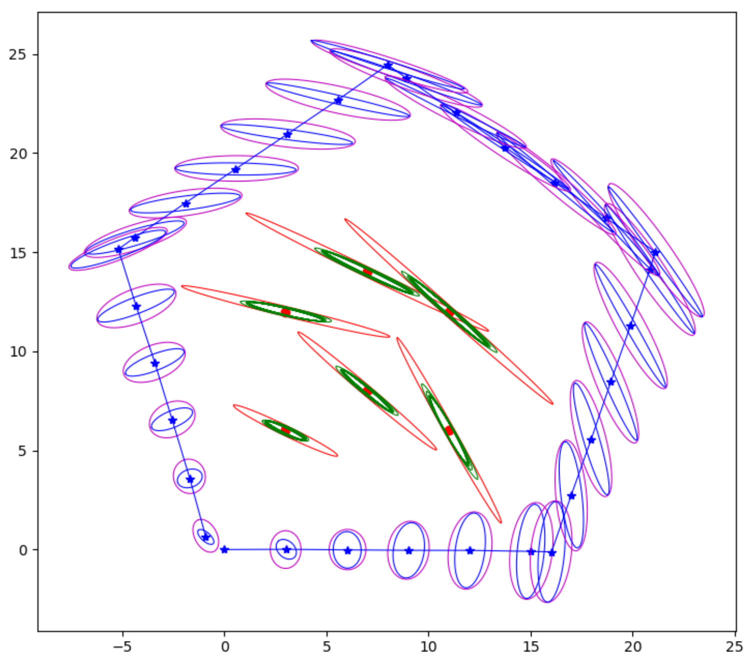

Figure 6: Visualization result with $10 \times \sigma_\beta$

When we increase the σ_r by 10 times, the result shown in Figure 7. The mapping is influenced a lot as the uncertainty of the landmark positions becomes much larger. Both predicted and updated uncertainty of the robot poses are increased. It heavily affects the performance of both localization and mapping of the algorithm.

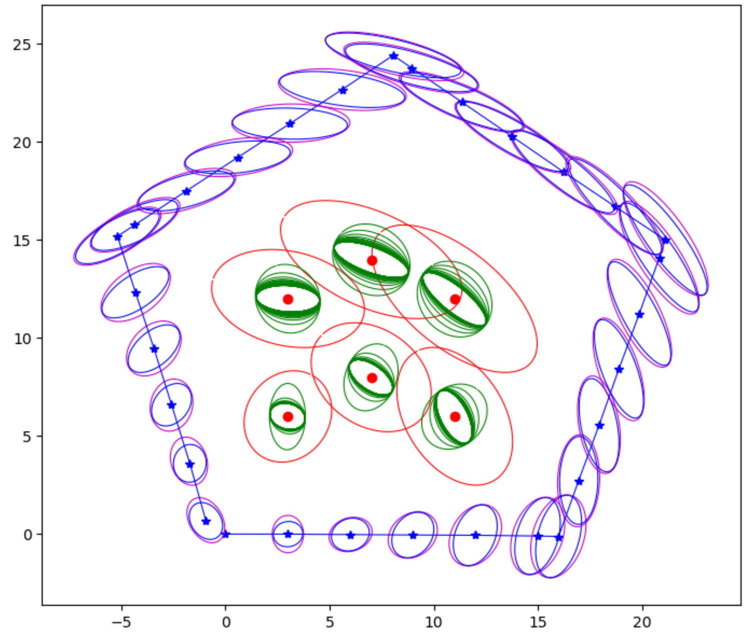

Figure 7: Visualization result with $10 \times \sigma_r$

3. The Compressed Extended Kalman Filter (CEKF) algorithm significantly reduces the computational requirement without introducing any penalties in the accuracy of the results. A CEKF stores and maintains all the information gathered in a local area with a cost proportional to the square of the number of landmarks in the area. This information is then transferred to the rest of the global map with a cost that is similar to full SLAM but in only one iteration. Similarly, we can initialize with much bigger matrix and use sparse matrix techniques to speed up the computation.

   A second approach would be to set a threshold or selection criteria to choose landmarks. For example, we may only consider those landmarks within a certain distance. We can also do range-only or bearing-only tracking for the robot to reduce the computational complexity.