Robot Localization and Mapping - Homework 2

Tianzhi Li
tianzhil@andrew.cmu.edu

March 13, 2025

1 Theory

1.1 State Transition Model

The state of the robot at time step t is represented as a vector:

$$\mathbf{p}_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} \tag{1}$$

where x_t and y_t are the global coordinates, and θ_t is the orientation.

Given the control inputs, the state evolves as follows:

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

This represents the robot moving d_t in the direction of its current heading θ_t and then rotating by an angle α_t.

1.2 Uncertainty Propagation

In reality, movement is affected by noise, which is assumed to follow a Gaussian distribution. The uncertainty in the robot's local coordinates is given by:

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

Since noise is defined in the robot's local frame, we must transform it into the global frame using a rotation matrix:

$$R(\theta_t) = \begin{bmatrix} \cos(\theta_t) & -\sin(\theta_t) & 0 \\ \sin(\theta_t) & \cos(\theta_t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

The covariance in the global frame is given by:

$$R_t^{\text{global}} = R(\theta_t) R_t R(\theta_t)^T \tag{5}$$

To propagate uncertainty over time, we use the Jacobian of the motion model:

$$G_t = \begin{bmatrix} 1 & 0 & -d_t \sin(\theta_t) \\ 0 & 1 & d_t \cos(\theta_t) \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

The uncertainty at the next time step is computed as:

$$\Sigma_{t+1} = G_t \Sigma_t G_t^T + R_t^{\text{global}} \tag{7}$$

This equation accounts for both the propagation of the previous uncertainty and the addition of new motion noise.

1.3 Landmark Position Estimation

Given a landmark observed at bearing β and range r, its global coordinates can be estimated as:

$$l_x = x_t + (r + n_r) \cos(\theta_t + \beta + n_\beta) \tag{8}$$

$$l_y = y_t + (r + n_r) \sin(\theta_t + \beta + n_\beta) \tag{9}$$

where n_r and n_β represent measurement noise in range and bearing, respectively.

1.4 Measurement Prediction

If the landmark position is known in global coordinates, the expected measurement is:

$$\hat{\beta} = warp2pi(np.arctan2(l_y - y_t, l_x - x_t) + n_\beta - \theta_t) \tag{10}$$

$$\hat{r} = \sqrt{(l_x - x_t)^2 + (l_y - y_t)^2} + n_r \tag{11}$$

This converts the landmark's absolute position back into relative bearing and range.

1.5 Jacobian with Respect to Robot Pose

The measurement Jacobian with respect to the robot pose is:

$$H_p = \begin{bmatrix} \frac{l_y - y_t}{(l_x - x_t)^2 + (l_y - y_t)^2} & -\frac{l_x - x_t}{(l_x - x_t)^2 + (l_y - y_t)^2} & -1 \\ -\frac{l_x - x_t}{\sqrt{(l_x - x_t)^2 + (l_y - y_t)^2}} & -\frac{l_y - y_t}{\sqrt{(l_x - x_t)^2 + (l_y - y_t)^2}} & 0 \end{bmatrix} \tag{12}$$

This matrix describes how the measurement changes with respect to the robots position.

1.6 Jacobian with Respect to Landmark Position

Similarly, the measurement Jacobian with respect to the landmark position is:

$$H_l = \begin{bmatrix} \frac{-(l_y - y_t)}{(l_x - x_t)^2 + (l_y - y_t)^2} & \frac{l_x - x_t}{(l_x - x_t)^2 + (l_y - y_t)^2} \\ \frac{l_x - x_t}{\sqrt{(l_x - x_t)^2 + (l_y - y_t)^2}} & \frac{l_y - y_t}{\sqrt{(l_x - x_t)^2 + (l_y - y_t)^2}} \end{bmatrix} \tag{13}$$

Since each landmark is independent, the measurement Jacobian is only computed for the observed landmark. This means H_l is only applied to the corresponding landmark, and other landmarks do not influence this computation.

By incorporating these Jacobians into the Extended Kalman Filter framework, the robot can update its belief about its position and the landmarks in the environment based on new sensor observations.

2 Implementation and Evaluation

2.1

The number of landmarks is 6.

2.2

Visualization is shown in Figure 1.

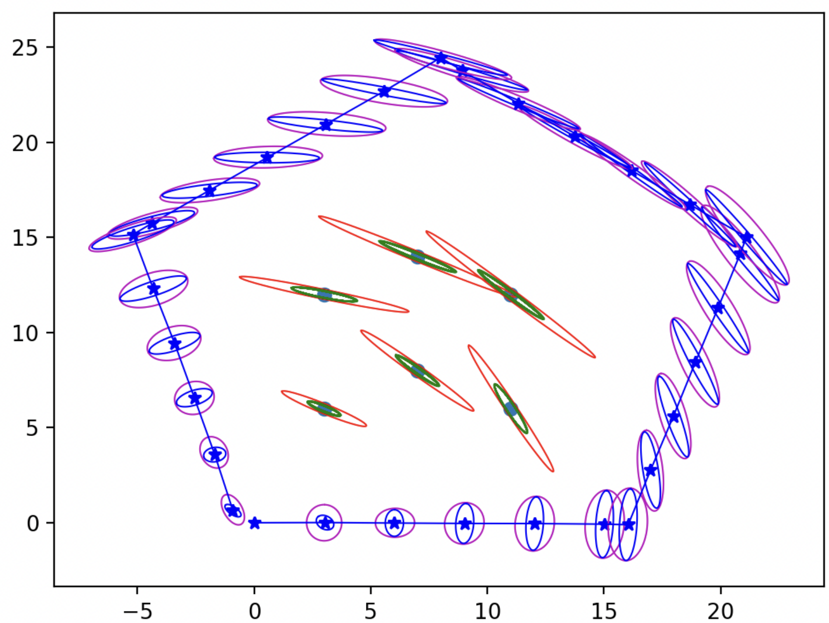


Figure 1: EKF Visualization

2.3

EKF-SLAM improves estimation by iteratively refining both the robots trajectory and the map. The magenta ellipses represent the predicted uncertainty, which increases due to motion noise, while the blue ellipses show the reduced uncertainty after incorporating sensor measurements, leading to improved localization. Similarly, newly detected landmarks have red ellipses, indicating high initial uncertainty, but as the robot revisits them, repeated observations reduce their uncertainty, resulting in smaller green ellipses. The consistent shrinkage of both robot and landmark uncertainty ellipses highlights how EKF-SLAM effectively balances prediction and update steps, ensuring a more accurate trajectory and a progressively refined map over time.

2.4

The ground truth positions of the landmarks have been plotted in Figure 2:

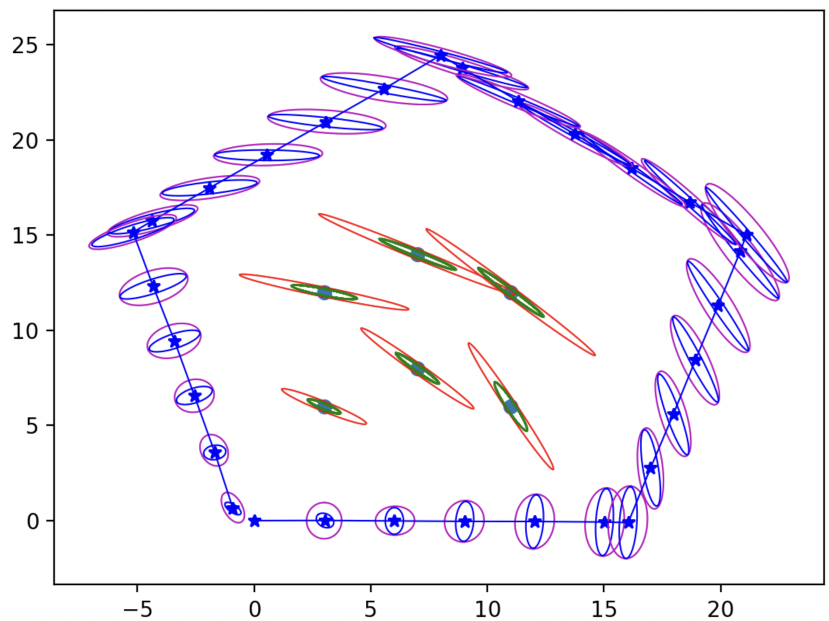


Figure 2: EKF Visualization with GroundTruth Landmarks

All ground truth positions lie within the smallest corresponding uncertainty ellipses. This indicates that the estimated landmark positions are consistent with the ground truth, and the uncertainties (represented by the ellipses) accurately capture the true positions.

The Euclidean and Mahalanobis distances between the estimated landmark positions and their ground truth positions have been computed and are reported in the table below:

Landmark	**Euclidean Distance**	**Mahalanobis Distance**
0	0.0027	0.0524
1	0.0034	0.0622
2	0.0048	0.0373
3	0.0054	0.0643
4	0.0043	0.0251
5	0.0047	0.0954

Table 1: Euclidean and Mahalanobis Distances for Landmarks

Euclidean Distance measures the straight-line distance between the estimated and ground truth positions. The small values (e.g., 0.0027 for Landmark 0) indicate that the estimated positions are very close to the ground truth.

Mahalanobis Distance measures the distance normalized by the uncertainty (covariance) of the estimate. The values are also small (e.g., 0.0524 for Landmark 0), indicating that the estimates are well within the expected uncertainty bounds.

3 Discussion

3.1

The initial covariance matrix is shown in Table 2.

0.0004	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0004	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0100	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.3651	-0.1791	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	-0.1791	0.0963	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	1.4579	-0.3630	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	-0.3630	0.0976	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.6492	-0.5621	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-0.5621	0.4986	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.9809	-0.9872	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-0.9872	0.5004	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.3698	-0.6653	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-0.6653	1.2262	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.4584	-1.3305
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-1.3305	1.2263

Table 2: Initial Covariance Matrix

The final covariance matrix is shown in Table 3.

0.0132	-0.0084	0.0004	0.0095	-0.0057	0.0097	-0.0056	0.0095	-0.0059	0.0098	-0.0059	0.0093	-0.0062	0.0097	-0.0061
-0.0084	0.0120	-0.0026	0.0062	-0.0004	0.0198	-0.0006	0.0108	-0.0094	0.0243	-0.0095	0.0064	-0.0183	0.0198	-0.0184
0.0004	-0.0026	0.0016	-0.0084	0.0039	-0.0179	0.0040	-0.0116	0.0103	-0.0211	0.0103	-0.0084	0.0166	-0.0179	0.0166
0.0095	0.0062	-0.0084	0.0543	-0.0266	0.1046	-0.0266	0.0708	-0.0603	0.1214	-0.0603	0.0539	-0.0940	0.1045	-0.0941
-0.0057	-0.0004	0.0039	-0.0266	0.0151	-0.0503	0.0148	-0.0344	0.0307	-0.0581	0.0306	-0.0265	0.0465	-0.0502	0.0465
0.0097	0.0198	-0.0179	0.1046	-0.0503	0.2127	-0.0504	0.1405	-0.1221	0.2483	-0.1222	0.1047	-0.1939	0.2124	-0.1940
-0.0056	-0.0006	0.0040	-0.0266	0.0148	-0.0504	0.0151	-0.0345	0.0308	-0.0584	0.0307	-0.0265	0.0467	-0.0504	0.0467
0.0095	0.0108	-0.0116	0.0708	-0.0344	0.1405	-0.0345	0.0943	-0.0809	0.1637	-0.0810	0.0708	-0.1274	0.1405	-0.1274
-0.0059	-0.0094	0.0103	-0.0603	0.0307	-0.1221	0.0308	-0.0809	0.0721	-0.1426	0.0719	-0.0604	0.1130	-0.1220	0.1130
0.0098	0.0243	-0.0211	0.1214	-0.0581	0.2483	-0.0584	0.1637	-0.1426	0.2908	-0.1428	0.1216	-0.2271	0.2482	-0.2273
-0.0059	-0.0095	0.0103	-0.0603	0.0306	-0.1222	0.0307	-0.0810	0.0719	-0.1428	0.0722	-0.0604	0.1131	-0.1222	0.1132
0.0093	0.0064	-0.0084	0.0539	-0.0265	0.1047	-0.0265	0.0708	-0.0604	0.1216	-0.0604	0.0542	-0.0942	0.1046	-0.0942
-0.0062	-0.0183	0.0166	-0.0940	0.0465	-0.1939	0.0467	-0.1274	0.1130	-0.2271	0.1131	-0.0942	0.1798	-0.1939	0.1796
0.0097	0.0198	-0.0179	0.1045	-0.0502	0.2124	-0.0504	0.1405	-0.1220	0.2482	-0.1222	0.1046	-0.1939	0.2126	-0.1940
-0.0061	-0.0184	0.0166	-0.0941	0.0465	-0.1940	0.0467	-0.1274	0.1130	-0.2273	0.1132	-0.0942	0.1796	-0.1940	0.1800

Table 3: Final Covariance Matrix

Initially, the landmark covariance matrix contains zero terms because, at the start, no correlations between different landmarks or between the robot and the landmarks have been explicitly introduced. However, as the prediction and update steps progress, the Extended Kalman Filter (EKF) propagates uncertainties. The state transition matrix F and measurement Jacobians H influence how uncertainty spreads. Specifically, the motion model introduces uncertainty in robot motion, which propagates to the landmark estimates due to their dependence on the robot's pose. The measurement model updates both the robot's state and landmarks together, establishing statistical dependencies. The covariance matrix P is updated in the update step as $P = (I - KH)P_{\text{pre}}$. Since $(I - KH)$ is not necessarily diagonal, off-diagonal elements appear in P, introducing correlations between previously uncorrelated state variables. These correlations arise because each update step introduces new information that links different state components. This results in non-zero correlations in the covariance matrix.

One of the assumptions is that the initial pose-landmark covariance is zero, meaning that landmarks are independent of the robots pose. However, this is not necessarily correct because if landmarks are estimated relative to the robots pose, their uncertainties should be correlated with the pose. The robots pose uncertainty directly affects the accuracy of landmark estimates, as landmarks are often observed and localized based on the robots position and orientation.

Another assumption is the initial landmark-landmark covariance is zero. This implies that landmarks are independent of each other. However, this assumption may not hold in practice. If landmarks are measured from the same pose, their estimates should be weakly correlated due to common motion and observation noise. For example, errors in the robots pose or sensor noise during observation can introduce shared uncertainties among landmarks observed simultaneously.

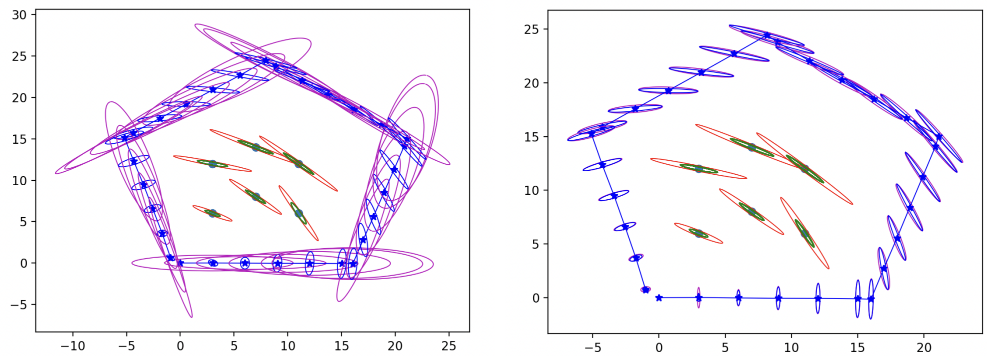
3.2

3.2.1 How σ_x influences the results

The parameter σ_x represents the standard deviation of the noise in the robot's motion model, specifically in the x-direction. It influences how much uncertainty is introduced into the robot's pose estimation during the prediction step of the EKF-SLAM algorithm. The visualization results are shown in Figure 3 and σ_x determines the spread of the ellipse along the x-axis. A low σ_x (e.g., 0.025) results in small uncertainty ellipses and precise estimates but risks overconfidence if actual noise is higher. A moderate σ_x (e.g., 0.25) balances uncertainty and precision, yielding reliable estimates. A high σ_x (e.g., 2.5) introduces large uncertainty ellipses, which makes the filter conservative and robust to high noise but potentially less precise if noise is overestimated.

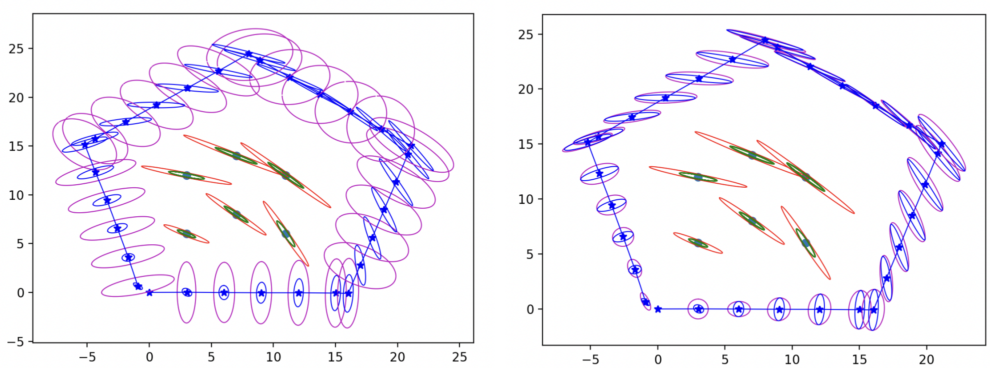
3.2.2 How σ_y influences the results

The parameter $\sigma_y$represents the standard deviation of the noise in the robot's motion model, specifically in the y-direction. It influences how much uncertainty is introduced into the robot's pose estimation during the prediction step of the EKF-SLAM algorithm. The visualization results are shown in Figure 4, and σ_y determines the spread of the ellipse along the y-axis.



(a) $\sigma_x = 2.5$ (b) $\sigma_x = 0.025$

Figure 3: Modify σ_x

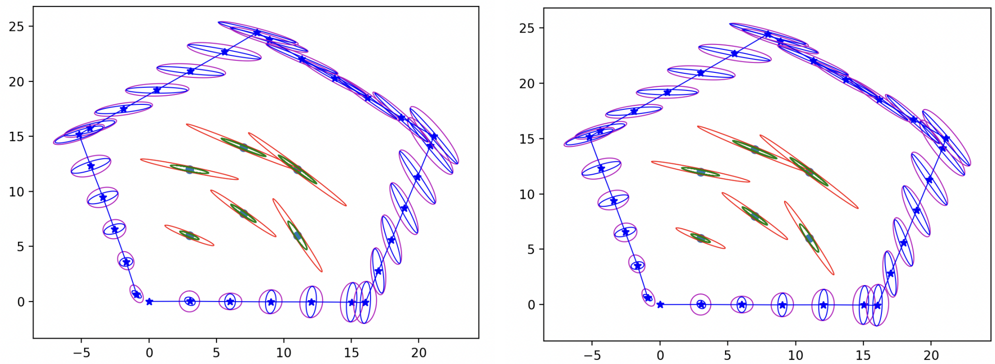


(a) $\sigma_y = 1$ (b) $\sigma_y = 0.01$

Figure 4: Modify σ_y

3.2.3 How σ_α influences the results

The parameter σ_α represents the covariance of rotation noise in the robot's motion model, affecting the uncertainty in the robot's orientation. While σ_α directly influences orientation estimates, its impact on the covariance matrix (which primarily encodes position information) can be subtle. This is because errors in orientation propagate into position estimates, but the effect may not always be visually prominent in the uncertainty ellipses, especially if other noise sources (e.g., σ_x or σ_y) dominate. For low σ_α (e.g., 0.01), orientation uncertainty is minimal, leading to precise but potentially overconfident estimates. For moderate σ_α (e.g., 0.1), orientation errors are balanced, contributing to reliable position estimates. For high σ_α (e.g., 1), large orientation uncertainty can propagate into position estimates, but the effect may not significantly alter the shape or size of the ellipses if position noise dominates. Thus, while σ_α influences the system, its impact on the visualization may be less pronounced compared to σ_x or σ_y.

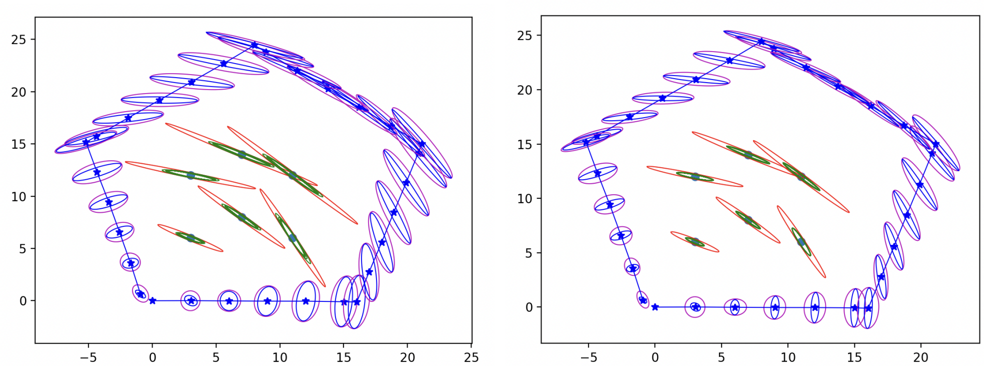


(a) $\sigma_\alpha = 1$ (b) $\sigma_\alpha = 0.01$

Figure 5: Modify σ_α

3.2.4 How σ_β influences the results

The parameter σ_β represents the covariance of observation noise in the measurement model, affecting the uncertainty in landmark estimates. It directly influences the update step of the EKF-SLAM algorithm by determining how much weight is given to new observations relative to the current state estimate. For low σ_β(e.g., 0.001), observation noise is minimal, leading to high confidence in measurements. This results in smaller uncertainty ellipses and precise landmark estimates, but the filter may become overconfident if the actual observation noise is higher. For moderate σ_β (e.g., 0.01), observation noise is balanced, allowing the filter to incorporate new measurements while maintaining reasonable uncertainty in landmark estimates. This yields reliable and robust results. For high σ_β (e.g., 0.1), observation noise is significant, leading to larger uncertainty ellipses and less precise landmark estimates. While this makes the filter more conservative, it ensures robustness to high observation noise.

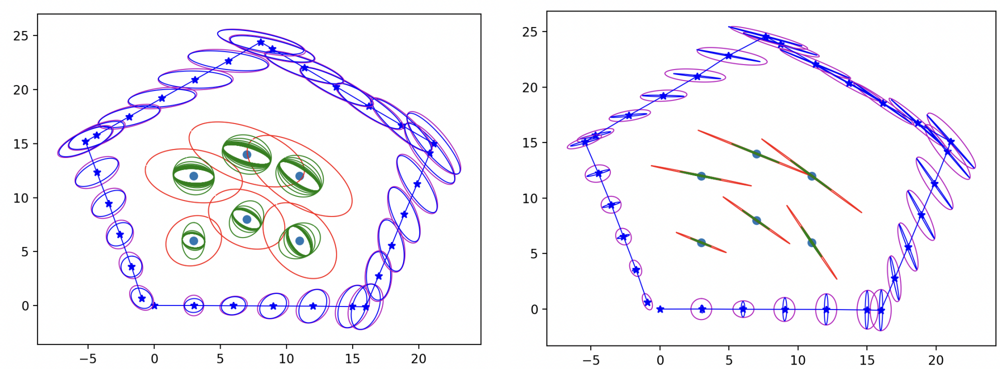


(a) $\sigma_\beta = 0.1$ (b) $\sigma_\beta = 0.001$

Figure 6: Modify σ_β

3.2.5 How σ_r influences the results

The parameter σ_r represents the covariance of range noise in the measurement model, affecting the uncertainty in landmark distance estimates. It directly influences the update step of the EKF-SLAM algorithm by determining how much weight is given to range measurements relative to the current state estimate. For low σ_r (e.g., 0.008), range noise is minimal, leading to high confidence in measurements. This results in smaller uncertainty ellipses and precise landmark estimates, but the filter may become overconfident if the actual range noise is higher. For moderate σ_r (e.g., 0.08), range noise is balanced, allowing the filter to incorporate new measurements while maintaining reasonable uncertainty in landmark estimates. This yields reliable and robust results. For high σ_r (e.g., 0.8), range noise is significant, leading to larger uncertainty ellipses and less precise landmark estimates. While this makes the filter more conservative, it ensures robustness to high range noise.



(a) $\sigma_r = 0.8$ (b) $\sigma_r = 0.008$

Figure 7: Modify σ_r

3.3

As the number of landmarks increases, the standard EKF-SLAM framework becomes computationally expensive due to the growth of the state vector and covariance matrix. To maintain constant-time complexity per iteration, the following optimizations can be applied:

1. Maintain and update only a subset of landmarks within the robot's vicinity instead of all detected landmarks.

2. Exploit sparsity by using efficient storage and computation techniques, such as sparse matrix representations, to reduce unnecessary computations.

3. Remove distant or less frequently observed landmarks while preserving their influence on the state through marginalization.

4. Divide the environment into overlapping submaps and perform local SLAM within each submap before merging the results globally.