16-833: Robot Localization and Mapping, Fall 2024

Solution 2 - SLAM using Extended Kalman Filter (EKF-SLAM)

Discussed with Ishir (iroongta) and Naman (nmenezes)

1 Theory (40 points)

Solution 1.1: The non-linear function for the next pose is,

$$p_{t+1} = g(p_t, u_t)$$

The non-linear function g(x) can be further elaborated as,

$$x_{t+1} = x_t + d_t \cos(\theta_t)$$
$$y_{t+1} = y_t + d_t \sin(\theta_t)$$
$$\theta_{t+1} = \theta_t + \alpha_t$$

where $\mathbf{p}_t = (x_t, y_t, \theta_t)$ is the 2D coordinates of the robot at time t.

Solution 1.2: The predicted uncertainty at $t + 1$ $\mathcal{N}(0, \Sigma_{t+1})$ is written as,

$$\Sigma_{t+1} = G\Sigma_t G^T + FR_tF^T$$

where Σ_t is the covariance in the current robot's pose

Jacobian G measures the effect of uncertainty in the current pose translated to the next pose and can be calculated as,

$$G = \begin{bmatrix} \frac{\partial g_1(p_t,u_t)}{\partial x} & \frac{\partial g_1(p_t,u_t)}{\partial y} & \frac{\partial g_1(p_t,u_t)}{\partial \theta} \\ \frac{\partial g_2(p_t,u_t)}{\partial x} & \frac{\partial g_2(p_t,u_t)}{\partial y} & \frac{\partial g_2(p_t,u_t)}{\partial \theta} \\ \frac{\partial g_3(p_t,u_t)}{\partial x} & \frac{\partial g_3(p_t,u_t)}{\partial y} & \frac{\partial g_3(p_t,u_t)}{\partial \theta} \end{bmatrix}$$

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

$$R_t = \begin{bmatrix} \sigma_x^2 & 0 & 0) \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix}$$

Since errors, R_t is in the local-body frame(as per the given image) but the pose is in the global coordinate frame, we need to calculate the F matrix. Adding noise to the equation in 1.1,

$$p_{t+1} = g(p_t, u_t) + \epsilon$$

Converting, noise in the global co-ordinate frame, the final equations will look like this,

$$x_{t+1} = x_t + d_t \cos(\theta_t) + \epsilon_x \cos(\theta_t) - \epsilon_y \sin(\theta_t)$$
$$y_{t+1} = y_t + d_t \sin(\theta_t) + \epsilon_x \sin(\theta_t) + \epsilon_y \cos(\theta_t)$$
$$\theta_{t+1} = \theta_t + \alpha_t + \epsilon_\alpha$$

Calculating the Jacobian F by similar practise as G, we get

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

We can finally write the entire equation as,

$$\Sigma_{t+1} = \begin{bmatrix} 1 & 0 & -d_t \sin(\theta_t) \\ 0 & 1 & d_t \cos(\theta_t) \\ 0 & 0 & 1 \end{bmatrix} \Sigma_t \begin{bmatrix} 1 & 0 & -d_t \sin(\theta_t) \\ 0 & 1 & d_t \cos(\theta_t) \\ 0 & 0 & 1 \end{bmatrix}^T +$$
$$\begin{bmatrix} \cos(\theta_t) & -\sin(\theta_t) & 0 \\ \sin(\theta_t) & \cos(\theta_t) & 0 \\ 0 & 0 & 1 \end{bmatrix} R_t \begin{bmatrix} \cos(\theta_t) & -\sin(\theta_t) & 0 \\ \sin(\theta_t) & \cos(\theta_t) & 0 \\ 0 & 0 & 1 \end{bmatrix}^T$$

Solution 1.3: The estimated position (l_x, l_y) of landmark l in global coordinates as a function of $\mathbf{p}_t$, β, r, can be written as,

$$l_x = x_t + (r + n_r) \cos(\theta_t + \beta + n_\beta)$$
$$l_y = y_t + (r + n_r) \sin(\theta_t + \beta + n_\beta)$$

n_r and n_β are the noise terms in range and bearing angle respectively.
The equation also assumes that the β bearing angle is with respect to the current heading angle θ

Solution 1.4: From 1.3, we have the two equations for l_x and l_y and we need to find the estimated bearing range $\hat{\beta}$ and estimated range $\hat{r}$. Using basic trigonometry, we can find them as

$$\hat{\beta} = np.arctan2\left(\frac{l_y - y_t}{l_x - x_t}\right) - n_b - \theta_t$$
$$\hat{r} = \sqrt{(l_x - x_t)^2 + (l_y - y_t)^2} - n_r$$

n_r and n_β are the noise terms in range and bearing angle respectively.
x_t and y_t are current robot pose at t timestamp in x and y coordinates.

Solution 1.5: The measurement Jacobian H_p using the above equations with respect to robot pose is,

$$H_p = \begin{bmatrix} \frac{\partial \hat{\beta}}{\partial x_t} & \frac{\partial \hat{\beta}}{\partial y_t} & \frac{\partial \hat{\beta}}{\partial \theta_t} \\ \frac{\partial \hat{r}}{\partial x_t} & \frac{\partial \hat{r}}{\partial y_t} & \frac{\partial \hat{r}}{\partial \theta_t} \end{bmatrix}$$

Here, the first row is the partial derivative of the bearing angle, and the second row is the partial derivative of the range. This preserves the matrix for further use in code.

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

Solution 1.6: The measurement Jacobian H_l with respect to the corresponding landmark,

$$H_l = \begin{bmatrix} \frac{\partial \hat{\beta}}{\partial l_x} & \frac{\partial \hat{\beta}}{\partial l_y} \\ \frac{\partial \hat{r}}{\partial l_x} & \frac{\partial \hat{r}}{\partial l_y} \end{bmatrix}$$

Again, the first row is the partial derivative of the bearing angle, and the second row is the partial derivative of the range. This preserves the matrix for further use in code.

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

In the context of EKF SLAM, we only need to calculate the measurement Jacobian H_l with respect to the landmark l for the following reasons/assumptions:

- **Indipendent Measurements:** The measurements of different landmarks are treated as independent under the EKF formulation. Thus, changes in the pose related to one landmark do not affect the measurement model of another landmark directly.
- **Localization Assumption:** The EKF assumes that the measurement noise is Gaussian and that the robot observes one landmark at a time. Therefore, the only relevant Jacobian for the current measurement corresponds to the landmark being observed.
- **Linearization Around the Current State:** The EKF operates under the assumption that state estimates are linearized around the current state. When computing the Jacobian H_l, we consider how changes in the robot's pose affect the observed landmark's position. Other landmarks do not influence this specific measurement directly.

2 Implementation and Evaluation (45 points)

Solution 2.1: The fixed number of landmarks being observed over the entire sequence is 6.

Solution 2.2: Figure 1 presents the visualization after all timesteps have been completed. As the robot collects more measurements while moving, the covariance ellipses representing the landmarks shrink, indicating a reduction in uncertainty.

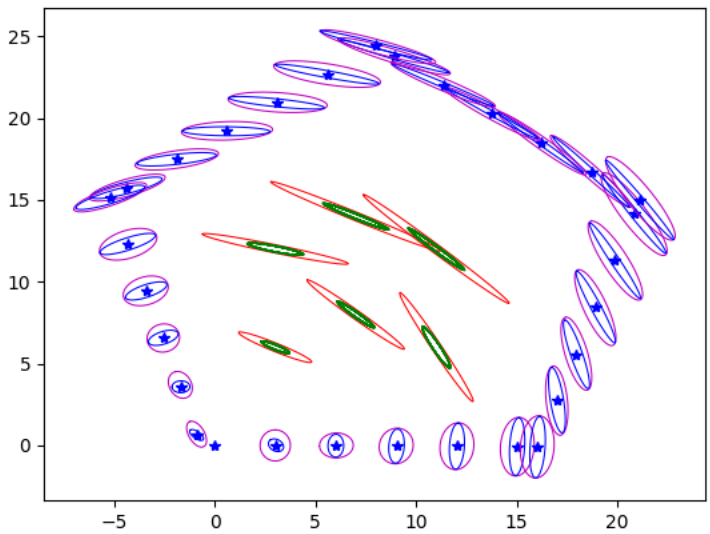

Figure 1: Visualization Plot after 29th Timestep

Additionally, as the robot moves, the update step reduces the covariance of the pose predicted in the preceding step.

Solution 2.3: EKF-SLAM improves the estimation of both the robot's trajectory and the map by progressively reducing uncertainty as more sensor data is integrated. Initially, both the robot and the landmarks have high uncertainty, represented by large ellipses. For landmarks farther from the robot, the uncertainty is greater, as depicted in figure 2. As the robot moves and new sensor measurements are collected, the uncertainty in the robot's pose (magenta ellipses) shrinks after the update, as shown by smaller blue ellipses.

Similarly, landmark uncertainty (green ellipses) decreases over time as repeated observations refine their positions, as depicted in figure 3, 4 and 5. This joint estimation process improves both the map and the trajectory, as better landmark estimates also enhance the robot's pose estimation. EKF-SLAM uses the Kalman Gain to weigh new measurements, resulting in continuous improvement of both trajectory and map estimates.

By jointly estimating the robot's pose and landmark positions, EKF-SLAM reduces uncertainty in both, evident in the shrinking ellipses as the system converges toward accurate estimates.

Solution 2.4: Figure 7 shows the plot for the Ground Truth and Estimated Landmark locations after the entire trajectory is complete. Yes, the ground truth and the estimated landmark location both lie inside the smaller corresponding ellipse (as depicted in green). This shows that the estimated landmark locations have been correctly refined over time.

Figure 8 shows the Mahalanobis and Euclidean distances of each landmark estimation with respect to the ground truth. The Mahalanobis distance, which considers the uncertainty in the estimate, shows that the estimated positions are well within the predicted uncertainty. The Euclidean distance quantifies the direct error in position. Smaller Mahalanobis distances indicate that the estimated positions are well

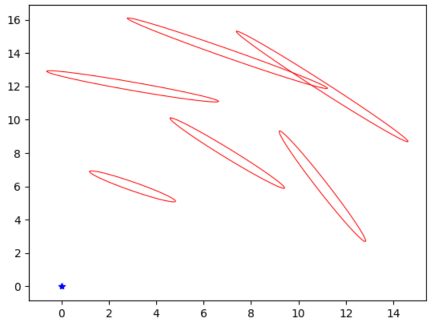

Figure 2: Initial Landmark Cov

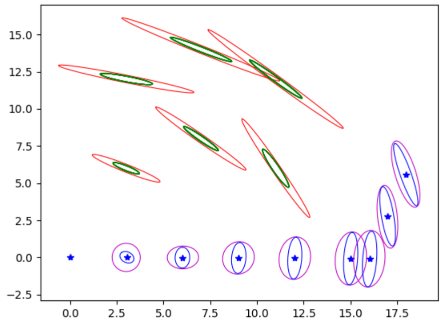

Figure 3: TimeStep 8

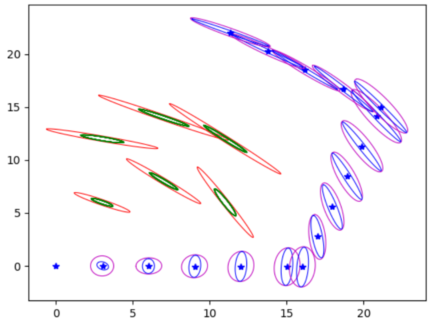

Figure 4: TimeStep 16

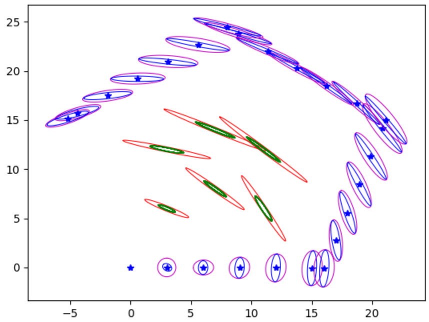

Figure 5: TimeStep 24

Figure 6: Visualization Plots from Start to 24th TimeStep

within the predicted uncertainty. Together, these numbers confirm that the estimation has progressively improved, with both distances decreasing over time, showing that the landmarks are getting closer to the ground truth as the uncertainty reduces.

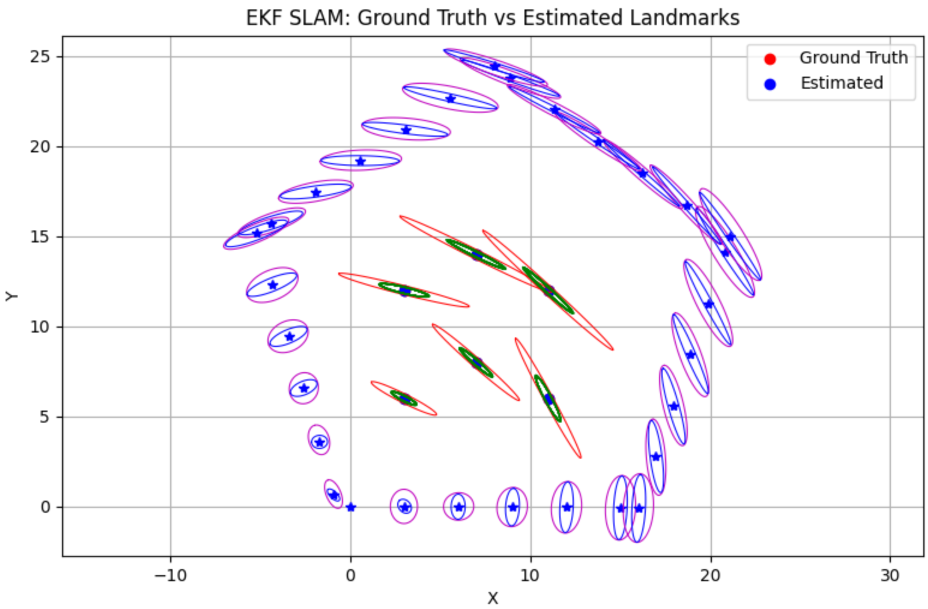

Figure 7: Ground Truth vs Estimated Landmarks EKF SLAM

```
Landmark :1
Euclidean distance: 0.0024213101875028313
Mahalanobis distance: 0.05332801663859403
Landmark :2
Euclidean distance: 0.006098549436755701
Mahalanobis distance: 0.06490181301470478
Landmark :3
Euclidean distance: 0.0013653618076648441
Mahalanobis distance: 0.03445168612526554
Landmark :4
Euclidean distance: 0.003596387513348785
Mahalanobis distance: 0.06480465067283431
Landmark :5
Euclidean distance: 0.002143562740557045 7
Mahalanobis distance: 0.022661149743450168
Landmark :6
Euclidean distance: 0.0060097119474222015
Mahalanobis distance: 0.09521298080532231
```

Figure 8: Mahalanobis and Euclidean Distances

3 Discussion (15 points)

Solution 3.1: Figure 9 shows the initial landmark covariance matrix and the figure 10 shows the full-state covariance matrix (pose and landmarks). The non-zero terms in the final state matrix for landmark covariances become non-zero over the course of the trajectory, because the robot's pose and variance affect the landmark estimate, as the update step take these into account when covariances for the landmarks.

The assumption that the state variables (robot pose and landmark positions) are uncorrelated with one another initially is in-correct. In reality, this is not necessarily true. Cross-covariances between the robot's pose and the landmarks may exist even from the beginning, but by setting only the diagonal values, you are effectively assuming that there are no initial correlations between them.

```
[[ 0.36505949 -0.17910678  0.          0.          0.          0.          0.          0.          0.          0.          0.          0.        ]
 [-0.17910678  0.09634191  0.          0.          0.          0.          0.          0.          0.          0.          0.          0.        ]
 [ 0.          0.          1.45788688 -0.36295314  0.          0.          0.          0.          0.          0.          0.          0.        ]
 [ 0.          0.         -0.36295314  0.09758366  0.          0.          0.          0.          0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.          0.64923055 -0.56208854  0.          0.          0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.         -0.56208854  0.49859405  0.          0.          0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.          0.          0.          1.98088641 -0.98716342  0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.          0.          0.         -0.98716342  0.50044183  0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.          0.          0.          0.          0.          0.36984691 -0.66534339  0.          0.        ]
 [ 0.          0.          0.          0.          0.          0.          0.          0.         -0.66534339  1.22615158  0.          0.        ]
 [ 0.          0.          0.          0.          0.          0.          0.          0.          0.          0.          1.4583545  -1.33045555]
 [ 0.          0.          0.          0.          0.          0.          0.          0.          0.          0.         -1.33045555  1.22625954]]
```

Figure 9: Initial Landmark Covariance

```
Final Pose and Landmark Covariances
[[ 0.01319209 -0.00848852  0.00042644  0.0092296  -0.00560167  0.00908788 -0.00545436  0.00908422 -0.00555556  0.00907036 -0.00549404  0.00900154 -0.00557826  0.00904327 -0.00554714]
 [-0.00848852  0.01238724 -0.00260366  0.00597524 -0.00033993  0.01924082 -0.00044474  0.01047728 -0.00911603  0.02363322 -0.00915999  0.00617524 -0.01785049  0.01927232 -0.01787717]
 [ 0.00042644 -0.00260366  0.00147258 -0.00742294  0.0034879  -0.01583338  0.00348868 -0.01023876  0.00907923 -0.01863698  0.00907694 -0.00745384  0.0146753  -0.01584041  0.01467324]
 [ 0.0092296   0.00597524 -0.00742294  0.04869967 -0.02403725  0.09276826 -0.02391348  0.06318914 -0.05356582  0.10755469 -0.05348453  0.0483832  -0.08315211  0.09273844 -0.08311072]
 [-0.00560167 -0.00033993  0.0034879  -0.02403725  0.01396405 -0.04487219  0.01361008 -0.03097005  0.02759954 -0.05183258  0.02752809 -0.02398185  0.04153973 -0.04486199  0.04148727]
 [ 0.00908788  0.01924082 -0.01583338  0.09276826 -0.04487219  0.18759309 -0.04475982  0.1243285  -0.10775346  0.21886863 -0.10769972  0.09292216 -0.17076483  0.18736539 -0.17073605]
 [-0.00545436 -0.00044474  0.00348868 -0.02391348  0.01361008 -0.04475982  0.01382254 -0.03086623  0.02751732 -0.05172221  0.02745282 -0.02387977  0.04145318 -0.04475648  0.04140333]
 [ 0.00908422  0.01047728 -0.01023876  0.06318914 -0.03097005  0.1243285  -0.03086623  0.08387358 -0.07167156  0.14472646 -0.07159973  0.06322272 -0.11243431  0.12433273 -0.11239802]
 [-0.00555556 -0.00911603  0.00907923 -0.05356582  0.02759954 -0.10775346  0.02751732 -0.07167156  0.06396497 -0.12585369  0.06361271 -0.05365405  0.09981425 -0.10779972  0.09977175]
 [ 0.00907036  0.02363322 -0.01863698  0.10755469 -0.05183258  0.21886863 -0.05172221  0.14472646 -0.12585369  0.25632581 -0.1258157   0.10777515 -0.20003677  0.21895849 -0.20001464]
 [-0.00549404 -0.00915999  0.00907694 -0.05348453  0.02752809 -0.10769972  0.02745282 -0.07159973  0.06361271 -0.1258157   0.06386293 -0.05357941  0.0997797  -0.10775616  0.09974378]
 [ 0.00900154  0.00617524 -0.00745384  0.0483832  -0.02398185  0.09292216 -0.02387977  0.06322272 -0.05365405  0.10777515 -0.05357941  0.04863752 -0.08335795  0.09290159 -0.08332541]
 [-0.00557826 -0.01785049  0.0146753  -0.08315211  0.04153973 -0.17076483  0.04145318 -0.11243431  0.09981425 -0.20003677  0.0997797  -0.08335795  0.15852324 -0.17086962  0.15820591]
 [ 0.00904327  0.01927232 -0.01584041  0.09273844 -0.04486199  0.18736539 -0.04475648  0.12433273 -0.10779972  0.21895849 -0.10775616  0.09290159 -0.17086962  0.1877238  -0.17084718]
 [-0.00554714 -0.01787717  0.01467324 -0.08311072  0.04148727 -0.17073605  0.04140333 -0.11239802  0.09977175 -0.20001464  0.09974378 -0.08332541  0.15820591 -0.17084718  0.15847137]]
```

Figure 10: Final State Covariance Matrix

Solution 3.2:

1. σ_x: This parameter represents the uncertainty in the robot's x direction (in the robot's frame, as shown in the images above). When the variance decreases,

the uncertainty in the x direction is reduced, resulting in smaller magenta ellipses in the x direction, as shown in Figure 11. When the given value of 0.25 is multiplied by 10, as shown in Figure 12, the uncertainty in the x direction increases significantly, causing the magenta ellipses to expand, indicating higher uncertainty in the x direction.

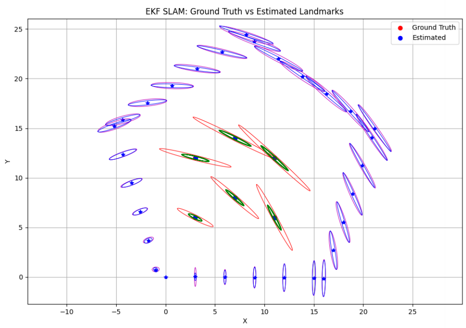

Figure 11: $\sigma_x = 0.5/10$

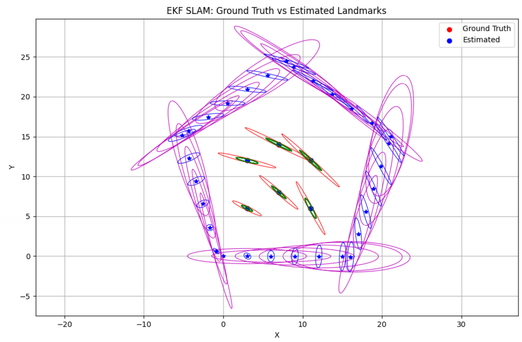

Figure 12: $\sigma_x = 0.25*10$

2. σ_y: This parameter represents the uncertainty in the robot's y direction (in the robot's frame, as shown in the images above). When the variance decreases, the uncertainty in the y direction is reduced, resulting in smaller magenta ellipses in the y direction, as shown in Figure 13. When the given value of 0.1 is multiplied by 10, as shown in Figure 14, the uncertainty in the y direction increases significantly, causing the magenta ellipses to expand, indicating higher uncertainty in the y direction.

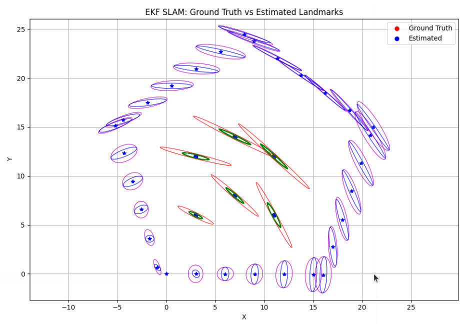

Figure 13: $\sigma_y = 0.1/10$

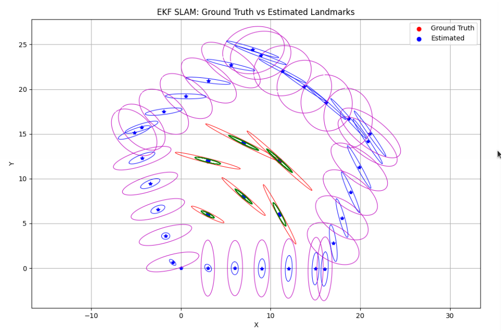

Figure 14: $\sigma_y = 0.1*10$

3. σ_α: This parameter represents the uncertainty in the robot's heading angle. When α value is increased from 0.1 to 1.0, the uncertainty in the robot's heading increases, which is also propagated to the estimated landmarks uncertainty. This can be seen by the slight increase and tilt in the ellipses of the landmarks (green) and robot (magenta) in the figure 16

4. σ_b: This parameter indicates the variance in the bearing angle of the measurement sensor. Increasing σ_b raises the sensor noise, resulting in larger ellipses for the estimated landmarks, as shown in figure 18. Additionally, the robot's pose uncertainty remains largely unchanged; as illustrated, the blue ellipses do not significantly shrink because there is noise in the sensor readings.

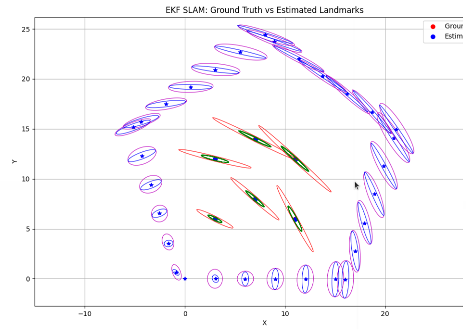

Figure 15: $\sigma_{alpha} = 0.1/10$

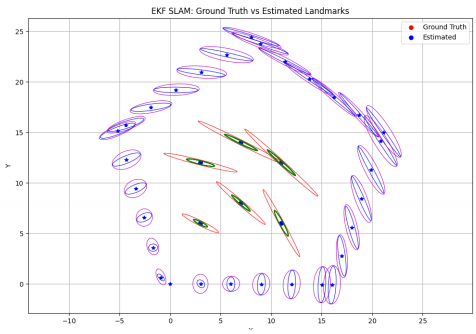

Figure 16: $\sigma_{alpha} = 0.1*10$

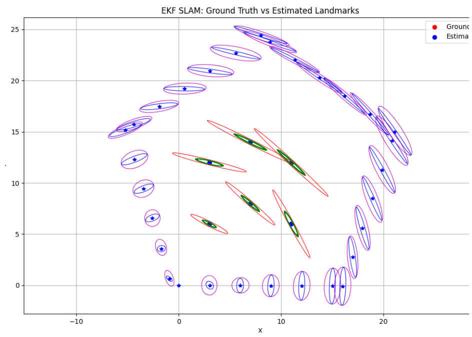

Figure 17: $\sigma_{beta} = 0.01/10$

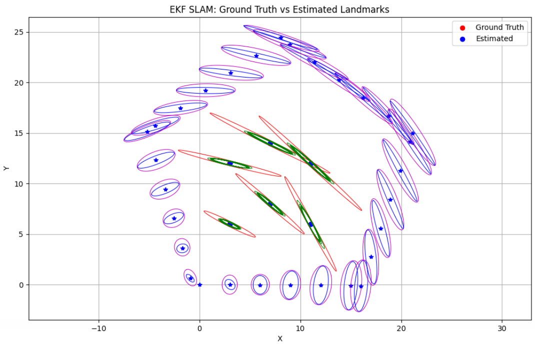

Figure 18: $\sigma_{beta} = 0.01*10$

5. σ_r: This parameter indicates the variance in the range of the measurement sensor. Increasing σ_b raises the sensor noise, resulting in larger ellipses(green) for the estimated landmarks, as shown in figure 20, they are significantly impacted.

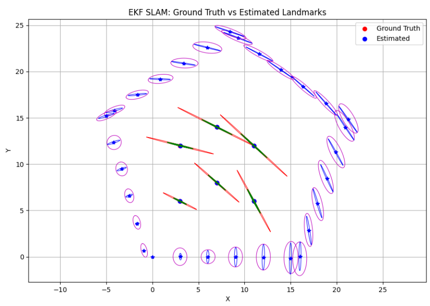

Figure 19: $\sigma_r = 0.08/10$

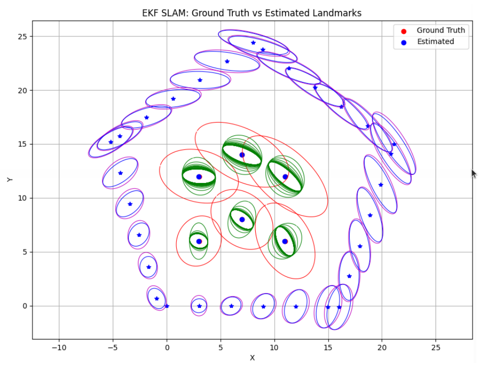

Figure 20: $\sigma_r = 0.08*10$

Solution 3.3:

1. **Landmark Management**: Implement a mechanism to manage landmarks by removing or merging redundant or less frequently observed landmarks. This reduces the overall number of landmarks that need to be processed.

2. **Submap Techniques**: Utilize submapping strategies where the environment is divided into smaller sections. The robot processes only a subset of landmarks

relevant to its current location, limiting the computations required at any given time.

3. **Sparse Covariance Representation**: Adopt a sparse representation for the covariance matrix to reduce memory usage and computational complexity. This can speed up the matrix operations needed during updates.

4. **Use of Local Maps**: Maintain a local map of nearby landmarks rather than a global map. This allows the EKF-SLAM system to focus on a smaller set of landmarks, reducing the computation load. This is also known as local