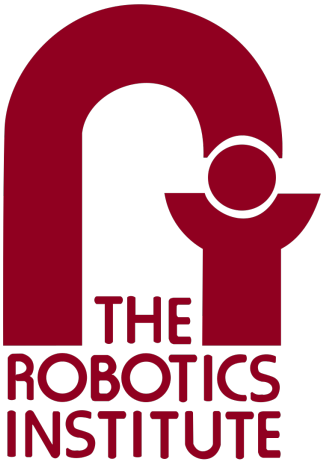
Homework 2



SLAM using Extended Kalman Filter (EKF-SLAM)

16-833A Spring 2025

Author: **Boxiang (William) Fu**
Andrew ID: boxiangf
E-mail: *boxiangf@andrew.cmu.edu*

February 16, 2025

Carnegie Mellon University
Robotics Institute

Contents

1 Theory

For this section, we define the state vector as $\mathbf{p} = \begin{bmatrix} x & y & \theta \end{bmatrix}^\top$, the control vector as $\mathbf{u} = \begin{bmatrix} d & \alpha \end{bmatrix}^\top$, and the measurement vector as $\mathbf{z} = \begin{bmatrix} r & \beta \end{bmatrix}^\top$. Similarly for the process and measurement noise, we denote their vectors as $\mathbf{w} = \begin{bmatrix} e_x & e_y & e_\alpha \end{bmatrix}^\top$ and $\mathbf{v} = \begin{bmatrix} n_r & n_\beta \end{bmatrix}^\top$ respectively. It is assumed that $\mathbf{w} \sim \mathcal{N}(0, \mathbf{Q})$ with $\mathbf{Q} = diag(\sigma_x^2, \sigma_y^2, \sigma_\alpha^2)$ and $\mathbf{v} \sim \mathcal{N}(0, \mathbf{R})$ with $\mathbf{R} = diag(\sigma_r^2, \sigma_\beta^2)$. Vectors and matricies with a subscript t denote the vector at time-step t.

1.1 Q1.

Assuming no noise or error in the control system, the next pose is given by the function $\mathbf{p}_{t+1} = f(\mathbf{p}_t, \mathbf{u}_t)$. From trigonometry (Figure 1), the change in pose in the global frame at time t is given by:

$$\begin{bmatrix} \Delta x_t \\ \Delta y_t \\ \Delta \theta_t \end{bmatrix} = \begin{bmatrix} d_t \cos(\theta_t) \\ d_t \sin(\theta_t) \\ \alpha_t \end{bmatrix}$$

With $\mathbf{p}_{t+1} = \Delta \mathbf{p}_t + \mathbf{p}_t$, the next pose $\mathbf{p}_{t+1}$ is given by:

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

Which is a non-linear function of the current pose and control inputs.

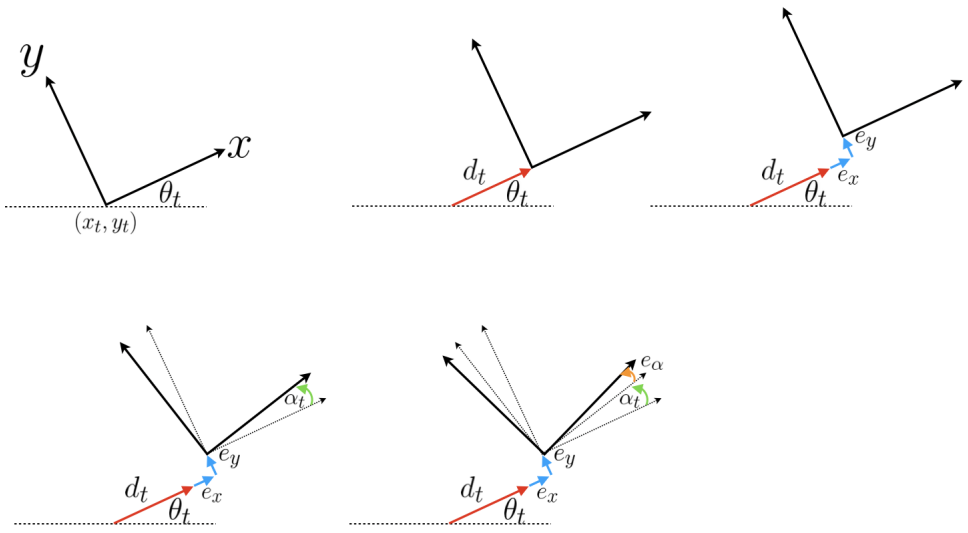


Figure 1: Planar Robot Kinematics

1.2 Q2.

Given the newfound error in the movement process, the change in pose in the global frame at time t is now given by:

$$\begin{bmatrix} \Delta x_t \\ \Delta y_t \\ \Delta \theta_t \end{bmatrix} = \begin{bmatrix} (d_t + e_x)\cos(\theta_t) - \sin(\theta_t) e_y \\ (d_t + e_x)\sin(\theta_t) + \cos(\theta_t) e_y \\ \alpha_t + e_\alpha \end{bmatrix}$$

The $-\sin(\theta_t)e_y$ and $+\cos(\theta_t)e_y$ term is easily derived using trigonometry as shown in Figure 2.

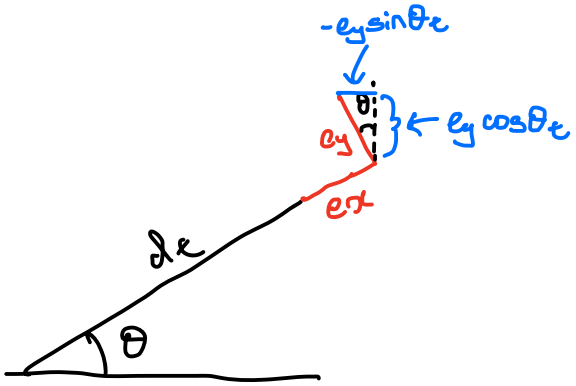


Figure 2: Derivation of Additional Terms in Pose Difference

With $\mathbf{p}_{t+1} = \Delta\mathbf{p}_t + \mathbf{p}_t$, the next pose $\mathbf{p}_{t+1}$ is given by:

$$\mathbf{p}_{t+1} = f(\mathbf{p}_t, \mathbf{u}_t, \mathbf{w}_t) = \begin{bmatrix} x_t + (d_t + e_x)\cos(\theta_t) - \sin(\theta_t)e_y \\ y_t + (d_t + e_x)\sin(\theta_t) + \cos(\theta_t)e_y \\ \theta_t + \alpha_t + e_\alpha \end{bmatrix}$$

Given that the covariance of the robot's pose at time t is given by $\mathbf{\Sigma}_t$, and assuming that the higher-order terms of the non-linearity in the movement process are negligible and can be ignored, we can obtain the covariance of the robot's pose at time $t+1$ as:

$$\mathbf{\Sigma}_{t+1} = \mathbf{F}_t\mathbf{\Sigma}_t\mathbf{F}_t^T + \mathbf{L}_t\mathbf{Q}_t\mathbf{L}_t^T$$

Where $\mathbf{Q}_t = diag(\sigma_x^2, \sigma_y^2, \sigma_\alpha^2)$, and $\mathbf{F}_t$ & $\mathbf{L}_t$ are Jacobian matrices given as follows:

$$\begin{aligned}
\mathbf{F}_t &= \left.\frac{\partial f}{\partial \mathbf{p}}\right|_{\mathbf{p}=\mathbf{p}_t,\mathbf{u}=\mathbf{u}_t,\mathbf{w}=0} \\
&= \left.\begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial \theta} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial \theta} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial \theta} \end{bmatrix}\right|_{\mathbf{p}=\mathbf{p}_t,\mathbf{u}=\mathbf{u}_t,\mathbf{w}=0} \\
&= \left.\begin{bmatrix} 1 & 0 & -(d+e_x)\sin(\theta) - \cos(\theta)e_y \\ 0 & 1 & (d+e_x)\cos(\theta) - \sin(\theta)e_y \\ 0 & 0 & 1 \end{bmatrix}\right|_{\mathbf{p}=\mathbf{p}_t,\mathbf{u}=\mathbf{u}_t,\mathbf{w}=0} \\
&= \begin{bmatrix} 1 & 0 & -d_t\sin(\theta_t) \\ 0 & 1 & d_t\cos(\theta_t) \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned}$$

And

$$
\begin{aligned}
\mathbf{L}_t &= \left.\frac{\partial f}{\partial \mathbf{w}}\right|_{\mathbf{p}=\mathbf{p}_t,\mathbf{u}=\mathbf{u}_t,\mathbf{w}=0} \\
&= \left.\begin{bmatrix} \frac{\partial f_1}{\partial e_x} & \frac{\partial f_1}{\partial e_y} & \frac{\partial f_1}{\partial e_\alpha} \\ \frac{\partial f_2}{\partial e_x} & \frac{\partial f_2}{\partial e_y} & \frac{\partial f_2}{\partial e_\alpha} \\ \frac{\partial f_3}{\partial e_x} & \frac{\partial f_3}{\partial e_y} & \frac{\partial f_3}{\partial e_\alpha} \end{bmatrix}\right|_{\mathbf{p}=\mathbf{p}_t,\mathbf{u}=\mathbf{u}_t,\mathbf{w}=0} \\
&= \left.\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}\right|_{\mathbf{p}=\mathbf{p}_t,\mathbf{u}=\mathbf{u}_t,\mathbf{w}=0} \\
&= \begin{bmatrix} \cos(\theta_t) & -\sin(\theta_t) & 0 \\ \sin(\theta_t) & \cos(\theta_t) & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

So the predicted uncertainty of the robot at time $t+1$ is represented by the 3-dimensional Gaussian distribution $\mathcal{N}(0, \mathbf{\Sigma}_{t+1})$, with $\mathbf{\Sigma}_{t+1}$ given as:

$$
\begin{aligned}
\mathbf{\Sigma}_{t+1} &= \mathbf{F}_t \mathbf{\Sigma}_t \mathbf{F}_t^T + \mathbf{L}_t \mathbf{Q}_t \mathbf{L}_t^T \\
&= \begin{bmatrix} 1 & 0 & -d_t \sin(\theta_t) \\ 0 & 1 & d_t \cos(\theta_t) \\ 0 & 0 & 1 \end{bmatrix} \cdot \mathbf{\Sigma}_t \cdot \begin{bmatrix} 1 & 0 & -d_t \sin(\theta_t) \\ 0 & 1 & d_t \cos(\theta_t) \\ 0 & 0 & 1 \end{bmatrix}^T \\
&+ \begin{bmatrix} \cos(\theta_t) & -\sin(\theta_t) & 0 \\ \sin(\theta_t) & \cos(\theta_t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\alpha^2 \end{bmatrix} \cdot \begin{bmatrix} \cos(\theta_t) & -\sin(\theta_t) & 0 \\ \sin(\theta_t) & \cos(\theta_t) & 0 \\ 0 & 0 & 1 \end{bmatrix}^T
\end{aligned}
$$

The mathematical derivation is attributed to Reference [3].

1.3 Q3.

Using the convention prescribed in Reference [2], we regard the bearing angle β and the range r as the measured values. Their values are related to the ground truth values by:

$$
\begin{aligned}
\beta &= \beta_{GroundTruth} + n_\beta \\
r &= r_{GroundTruth} + n_r
\end{aligned}
$$

The estimated position (l_x, l_y) of the landmark is measured with respect to the ground truth measurements of β and r. Referring to Figure 1, the location of the landmark in the robot's (odom) frame is given by:

$$
\begin{bmatrix} l_x^{odom} \\ l_y^{odom} \end{bmatrix} = \begin{bmatrix} (r - n_r)\cos(\beta - n_\beta) \\ (r - n_r)\sin(\beta - n_\beta) \end{bmatrix}
$$

The landmark in the global frame is given by a homogeneous transformation (rotation plus translation) from the robot frame to the global frame given by:

$$
\begin{aligned}
\begin{bmatrix} l_x \\ l_y \end{bmatrix} &= R \cdot \begin{bmatrix} l_x^{odom} \\ l_y^{odom} \end{bmatrix} + T \\
&= \begin{bmatrix} \cos(\theta_t) & -\sin(\theta_t) \\ \sin(\theta_t) & \cos(\theta_t) \end{bmatrix} \cdot \begin{bmatrix} (r - n_r)\cos(\beta - n_\beta) \\ (r - n_r)\sin(\beta - n_\beta) \end{bmatrix} + \begin{bmatrix} x_t \\ y_t \end{bmatrix} \\
&= \begin{bmatrix} \cos(\theta_t)(r - n_r)\cos(\beta - n_\beta) - \sin(\theta_t)(r - n_r)\sin(\beta - n_\beta) + x_t \\ \sin(\theta_t)(r - n_r)\cos(\beta - n_\beta) + \cos(\theta_t)(r - n_r)\sin(\beta - n_\beta) + y_t \end{bmatrix} \\
&= \begin{bmatrix} (r - n_r)\cos(\theta_t + \beta - n_\beta) + x_t \\ (r - n_r)\sin(\theta_t + \beta - n_\beta) + y_t \end{bmatrix}
\end{aligned}
$$

Where the rotation and translation transformations are the standard reference transforms in a 2D-plane and the last line comes from trigonometric identities [1]. Also note that the derivation above assumes the landmark's location is measured with respect to the ground truth bearing angle and range. Reference [2] also cites another possibility as using the estimated landmark's location as measured with respect to the actual bearing angle and range measured. This will flip the signs before n_r and n_β, but will have no actual effect on the EKF calculations (as we will see in Q5 and Q6) as the expectation of the noise is zero and the variance of the noise is agnostic of left- or right-directionality (since it is a squared value).

1.4 Q4.

Using the equality from Q3, we obtain:

$$
\begin{bmatrix} l_x - x_t \\ l_y - y_t \end{bmatrix} = \begin{bmatrix} (r - n_r)\cos(\theta_t + \beta - n_\beta) \\ (r - n_r)\sin(\theta_t + \beta - n_\beta) \end{bmatrix} \tag{1}
$$

Squaring both sides and adding them up, we obtain:

$$
(l_x - x_t)^2 + (l_y - y_t)^2 = (r - n_r)^2(\cos(\theta_t + \beta - n_\beta) + \sin(\theta_t + \beta - n_\beta))
$$

Noting the unit circle identity and taking the positive square root on both sides (since range cannot be negative), we obtain the range:

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

This time dividing the second row of Equation 1 by the first row, we obtain

$$
\frac{\sin(\theta_t + \beta - n_\beta)}{\cos(\theta_t + \beta - n_\beta)} = \frac{l_y - y_t}{l_x - x_t}
$$

Using the function *np.arctan2*(.) and warping *warp2pi*(.), we obtain the bearing:

$$
\beta = warp2pi(np.arctan2(l_y - y_t, l_x - x_t) - \theta_t + n_\beta) \tag{3}
$$

1.5 Q5.

The measurement model is given by the equation $\mathbf{z}_t = h(\mathbf{p}_t, \mathbf{l}, \mathbf{v}_t) = \begin{bmatrix} r & \beta \end{bmatrix}^\top$. The measurement Jacobian $\mathbf{H}_p$ is given by:

$$
\begin{aligned}
\mathbf{H}_p &= \left.\frac{\partial h}{\partial \mathbf{p}}\right|_{\mathbf{p}=\mathbf{p}_t,\mathbf{l}=\mathbf{l},\mathbf{v}=0} \\
&= \left.\begin{bmatrix} \frac{\partial h_1}{\partial x} & \frac{\partial h_1}{\partial y} & \frac{\partial h_1}{\partial \theta} \\ \frac{\partial h_2}{\partial x} & \frac{\partial h_2}{\partial y} & \frac{\partial h_2}{\partial \theta} \end{bmatrix}\right|_{\mathbf{p}=\mathbf{p}_t,\mathbf{l}=\mathbf{l},\mathbf{v}=0} \\
&= \left.\begin{bmatrix} \frac{\partial r}{\partial x} & \frac{\partial r}{\partial y} & \frac{\partial r}{\partial \theta} \\ \frac{\partial \beta}{\partial x} & \frac{\partial \beta}{\partial y} & \frac{\partial \beta}{\partial \theta} \end{bmatrix}\right|_{\mathbf{p}=\mathbf{p}_t,\mathbf{l}=\mathbf{l},\mathbf{v}=0}
\end{aligned}
$$

From Equations 2 and 3, calculating each in turn and evaluating at $\mathbf{p} = \mathbf{p}_t, \mathbf{l} = \mathbf{l}, \mathbf{v} = 0$:

$$
\begin{aligned}
\frac{\partial r}{\partial x} &= \frac{-(l_x - x_t)}{\sqrt{(l_x - x_t)^2 + (l_y - y_t)^2}} \\
\frac{\partial r}{\partial y} &= \frac{-(l_y - y_t)}{\sqrt{(l_x - x_t)^2 + (l_y - y_t)^2}} \\
\frac{\partial r}{\partial \theta} &= 0
\end{aligned}
$$

Noting that the derivative of $arctan(x)$ is $\frac{1}{1+x^2}$:

$$
\begin{aligned}
\frac{\partial \beta}{\partial x} &= \frac{1}{1 + (\frac{l_y - y_t}{l_x - x_t})^2} \cdot \frac{(l_y - y_t)}{(l_x - x_t)^2} \\
&= \frac{(l_x - x_t)^2}{(l_x - x_t)^2 + (l_y - y_t)^2} \cdot \frac{(l_y - y_t)}{(l_x - x_t)^2} \\
&= \frac{(l_y - y_t)}{(l_x - x_t)^2 + (l_y - y_t)^2} \\
\frac{\partial \beta}{\partial y} &= \frac{1}{1 + (\frac{l_y - y_t}{l_x - x_t})^2} \cdot \frac{-1}{(l_x - x_t)} \\
&= \frac{(l_x - x_t)^2}{(l_x - x_t)^2 + (l_y - y_t)^2} \cdot \frac{-1}{(l_x - x_t)} \\
&= -\frac{(l_x - x_t)}{(l_x - x_t)^2 + (l_y - y_t)^2} \\
\frac{\partial \beta}{\partial \theta} &= -1
\end{aligned}
$$

So the Jacobian is:

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

1.6 Q6.

The measurement model is given by the equation $\mathbf{z}_t = h(\mathbf{p}_t, \mathbf{l}, \mathbf{v}_t) = \begin{bmatrix} r & \beta \end{bmatrix}^\top$. The landmark Jacobian $\mathbf{H}_l$ is given by:

$$
\begin{aligned}
\mathbf{H}_l &= \left.\frac{\partial h}{\partial \mathbf{l}}\right|_{\mathbf{p}=\mathbf{p}_t,\mathbf{l}=\mathbf{l},\mathbf{v}=0} \\
&= \left.\begin{bmatrix} \frac{\partial h_1}{\partial l_x} & \frac{\partial h_1}{\partial l_y} \\ \frac{\partial h_2}{\partial l_x} & \frac{\partial h_2}{\partial l_y} \end{bmatrix}\right|_{\mathbf{p}=\mathbf{p}_t,\mathbf{l}=\mathbf{l},\mathbf{v}=0} \\
&= \left.\begin{bmatrix} \frac{\partial r}{\partial l_x} & \frac{\partial r}{\partial l_y} \\ \frac{\partial \beta}{\partial l_x} & \frac{\partial \beta}{\partial l_y} \end{bmatrix}\right|_{\mathbf{p}=\mathbf{p}_t,\mathbf{l}=\mathbf{l},\mathbf{v}=0}
\end{aligned}
$$

From Equations 2 and 3, calculating each in turn and evaluating at $\mathbf{p} = \mathbf{p}_t, \mathbf{l} = \mathbf{l}, \mathbf{v} = 0$:

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

Noting that the derivative of $arctan(x)$ is $\frac{1}{1+x^2}$:

$$
\begin{aligned}
\frac{\partial \beta}{\partial l_x} &= \frac{1}{1 + (\frac{l_y - y_t}{l_x - x_t})^2} \cdot \frac{-(l_y - y_t)}{(l_x - x_t)^2} \\
&= -\frac{(l_y - y_t)}{(l_x - x_t)^2 + (l_y - y_t)^2} \\
\frac{\partial \beta}{\partial l_y} &= \frac{1}{1 + (\frac{l_y - y_t}{l_x - x_t})^2} \cdot \frac{1}{(l_x - x_t)} \\
&= \frac{(l_x - x_t)}{(l_x - x_t)^2 + (l_y - y_t)^2}
\end{aligned}
$$

So the Jacobian is:

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

We do not need to calculate the measurement Jacobian with respect to other landmarks except for itself since we assumed the landmarks to be independent. This means that a change in another landmark does not influence a change in the range r or bearing angle β measured for the current landmark. In other words, the landmark Jacobian (i.e. rate of change) with respect to other landmarks will be the zero matrix.

2 Implementation and Evaluation

2.1 Q1.

The fixed number of landmarks being observed is **six**. This is because every odd data entry in `data/data.txt` is a measurement step and has 12 data points each. Since each landmark requires two data points (range r and bearing angle β), there is in total 6 corresponding landmarks.

2.2 Q2.

Using the default parameters shown in Table 1, the visualization of the trajectory and landmarks after all the steps are finished is shown in Figure 3.

Parameter	Value
σ_x	0.25
σ_y	0.1
σ_α	0.1
σ_β	0.01
σ_r	0.08

Table 1: Measurement and Motion Uncertainty Parameters

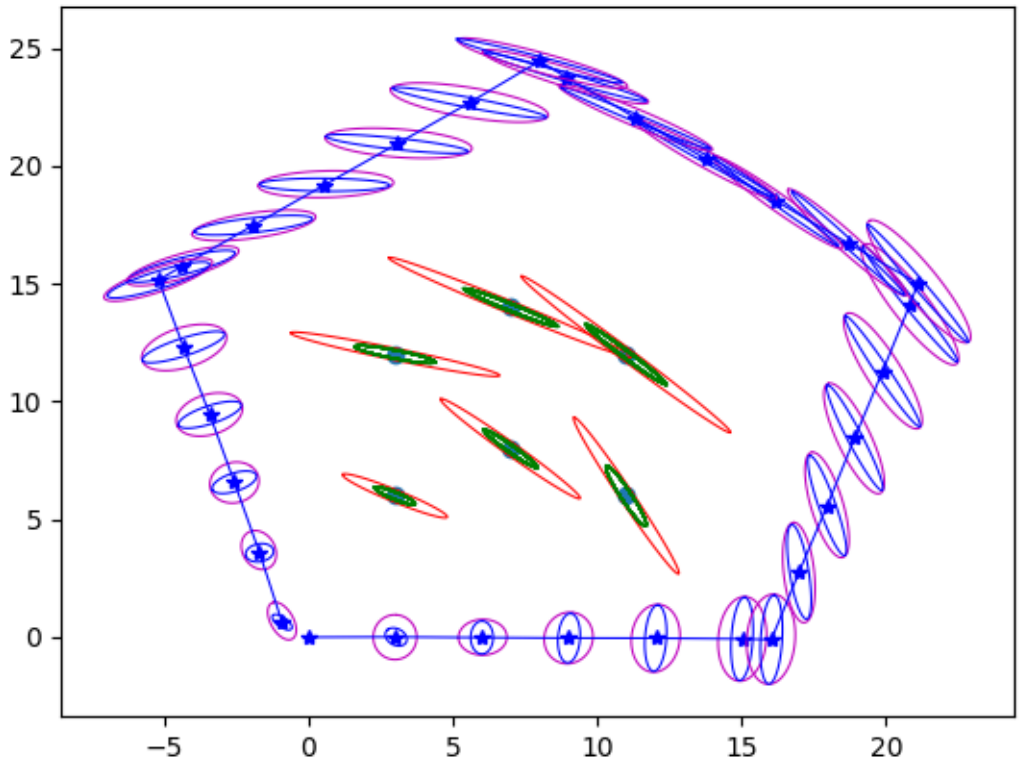


Figure 3: Trajectory Visualization Using EKF

2.3 Q3.

From Figure 3, we can see how the EKF-SLAM algorithm estimates the uncertainty of both the trajectory and map. During the predict step (magenta ellipses), the uncertainty increases due to noise from the motion model. This is expected since control inputs are not perfect nor are the movement of the robot, so noise inevitably causes the robot's state to increase in uncertainty. However, once additional information is provided in the update step (in the form of ray casting to known landmarks), the robot can correct the estimate and uncertainty in its pose by calculating the difference between the actual measurement and predicted measurement. This tends to decrease the pose uncertainty (blue ellipses) as we incorporate additional observation information. Finally, as the robot navigates back to its starting location, the pose uncertainty decreases dramatically. This is because the robot has essentially performed loop closure and is observing the landmarks at the same orientation again. This makes the robot more confident with its observations and thus a very low uncertainty in its trajectory estimation.

For the uncertainty in landmarks (map), there is originally high uncertainty (red ellipses) when only a few time-steps have passed. This is expected since we do not have much observational data to be sure of its locations. Furthermore, the further the landmark is, the more uncertain its position is. This is because we are estimating its location using ray projections, which has uncertainties that "fan out" as we increase distance. The landmarks obviously does not have a motion model as its location is assumed fixed, so its uncertainty does not increase during the predict step. However, the uncertainty decreases during the update step as we see the landmark from more locations and angles (green ellipses). Once the robot makes a revolution around the landmarks, the uncertainties reduce dramatically as we have observed the landmarks over 360 degrees and performs loop closure, thus increasing the estimate of the map.

In summary, the EKF-SLAM algorithm is a tug-of-war between increasing uncertainty due to motion (predict step), and decreasing uncertainty due to measurements (update step). As long as the algorithm is initialized such that the decrease in uncertainty dominates the increase in uncertainty, over time the algorithm will improve the estimate of both the trajectory and the map.

2.4 Q4.

The Euclidean and Mahalanobis distances of each landmark estimation with respect to the ground truth are tabulated in Table 2. The distances are very small, telling us that the estimated landmark locations are very close to the ground truth landmark locations. The Euclidean distance is the error in the mean of the posterior Normal distribution to the ground truth. The low values in Table 2 indicates that the centroid is well localized at the ground truth. The Mahalanobis distance is the error after taking into account of the covariances between the data points. The low values in Table 2 indicates that the centroid is again well localized at the ground truth, even after taking into account of covariance relationships.

Landmark	1	2	3	4	5	6
Euclidean Distance	0.0024	0.0058	0.0014	0.0033	0.0018	0.0056
Mahalanobis Distance	0.0545	0.0648	0.0351	0.0641	0.0220	0.0944

Table 2: Euclidean and Mahalanobis Distances for Each Landmark

The ground truth positions of the landmarks and the corresponding ellipses is plotted

in Figure 4. Each of the ground truth values are within the smallest corresponding ellipse. This means that our uncertainty estimates on the location of the landmarks are reliable and can accurately localize the landmarks' location. Additionally, the smallest ellipses quite tightly bound the ground truth. This indicates that the EKF-SLAM algorithm works well in reducing the uncertainty to only a small localized region centered near the ground truth.

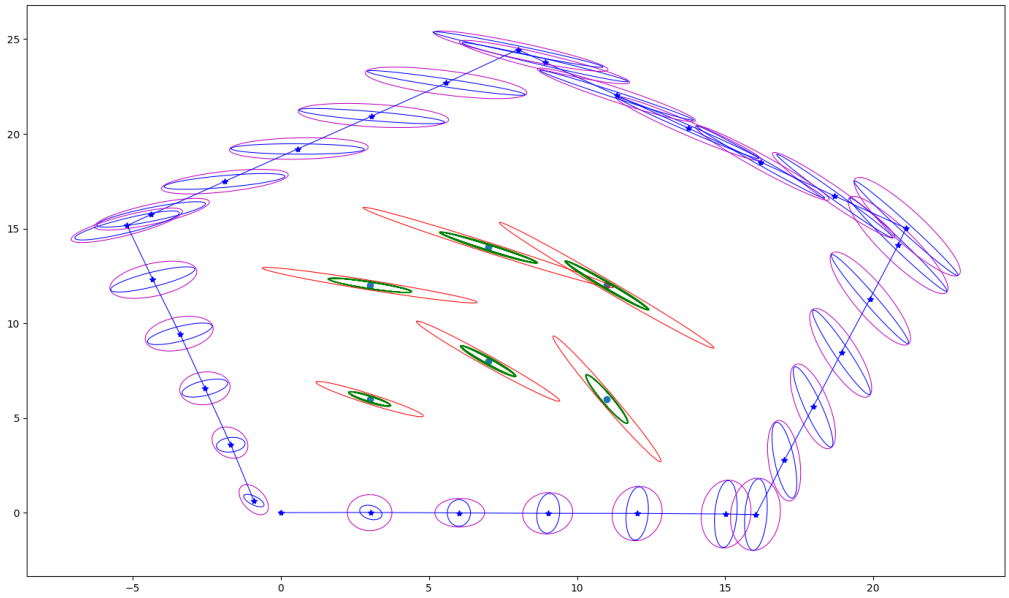


Figure 4: Ground Truth and Uncertainty Ellipse Visualization

3 Discussion

3.1 Q1.

The final state 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 P Matrix

Note that the off-diagonal blocks on the final P matrix is no longer zeros. From the EKF-SLAM algorithm, we note that the predict step does not change the covariance matrix of the landmarks (which is reasonable since there is no movement errors from the

stationary landmarks). Therefore, the only update to the covariance matrix of the landmarks is in the update step (when the landmarks gets observed and measured).

For the landmark-landmark covariances, observing multiple landmarks together induces correlations, thus making them no longer independent from each other. An extreme example is if one landmark is completely blocking the other. Then the observation of two landmarks are mutually exclusive, and thus not independent.

For the robot pose-landmark covariances, the measurement model couples pose and landmark uncertainties. Since the robot's pose is uncertain, and landmarks are estimated relative to it, their uncertainty grows correlated with the robot's pose uncertainty. Furthermore, if the pose-landmark covariances are uncorrelated, no information on the robot's pose would be obtained from measuring the landmarks. Therefore, the measurement model couples pose and landmark uncertainties to allow the robot to localize itself in the environment.

For both the landmark-landmark covariances and robot pose-landmark covariances, the mathematical step for when the covariances becomes correlated is during the Kalman update step $P = (I - KH)P_{pre}$. Less than perfect trust on the motion model will cause the KH term to be non-zero, thus adjusting the uncertainty and causing correlations between the robot's pose and landmarks since KH is not a block diagonal matrix.

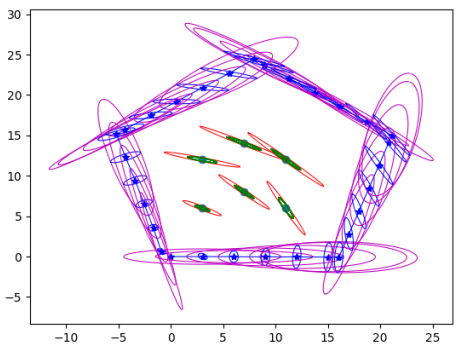
The initial assumption made regarding cross-covariances that is incorrect is that we assumed zero cross-covariances between the robot's pose and landmarks, as well as between landmarks. The landmark-landmark zero cross-covariance assumption is incorrect as the observation of one landmark certainly influences the observation of other landmarks (e.g. obstruction). The robot pose-landmark zero cross-covariance assumption is incorrect since the landmarks are estimated relative to the initial robot pose. If the robot's initial pose is uncertain, the landmarks should inherit this uncertainty, and thus be cross-correlated.

3.2 Q2.

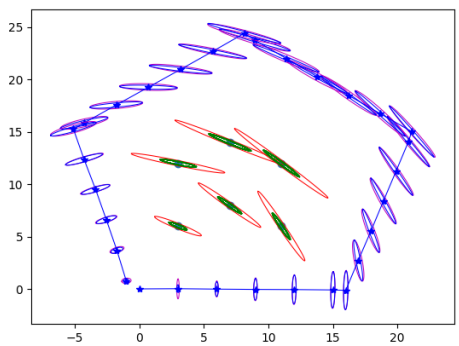
From the initial parameters in Table 1, we increase/decrease the parameters by a factor of 10, holding all others constant.

3.2.1 Increasing/Decreasing σ_x

Increasing/decreasing σ_x has the effect of increasing/decreasing the pose uncertainty in the robot's direction of travel. This is seen by the elongation of the covariance ellipse in the robot's direction of travel.



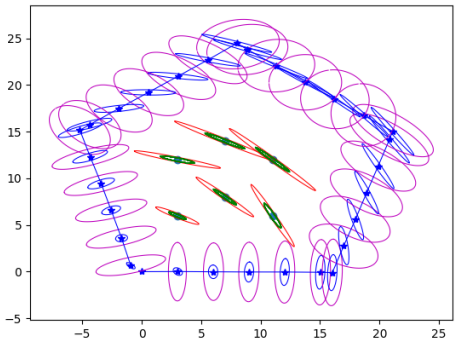
(a) Increasing σ_x by Factor of 10



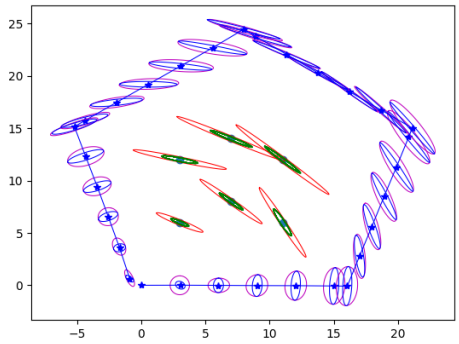
(b) Decreasing σ_x by Factor of 10

3.2.2 Increasing/Decreasing σ_y

Increasing/decreasing σ_y has the effect of increasing/decreasing the pose uncertainty in the direction perpendicular to the robot's direction of travel. This is seen by the elongation of the covariance ellipse in the robot's broadside.



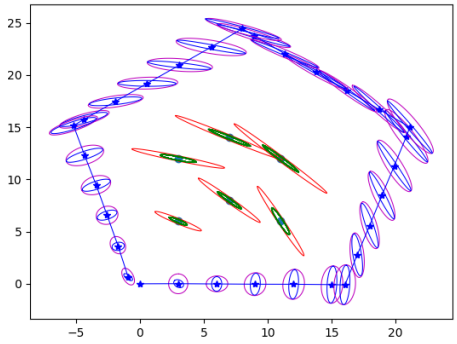
(a) Increasing σ_y by Factor of 10



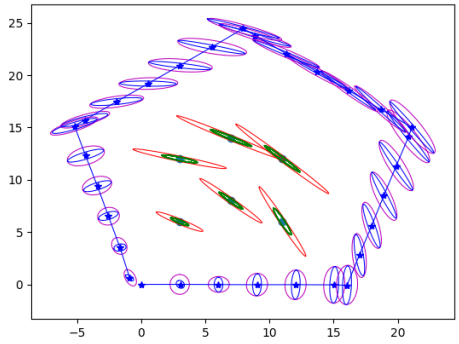
(b) Decreasing σ_y by Factor of 10

3.2.3 Increasing/Decreasing σ_α

Increasing/decreasing σ_α does not manifest itself in the figures. This is because only positional information is displayed, and not orientation. What we should expect from increasing/decreasing σ_α is the increase/decrease in pose uncertainty in the robot's orientation. This would manifest in expanding/contracting the covariance ellipse in the rotational state-space.



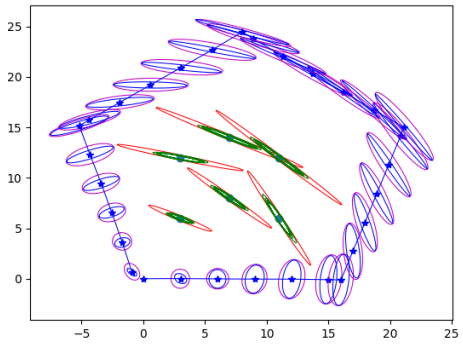
(a) Increasing σ_α by Factor of 10



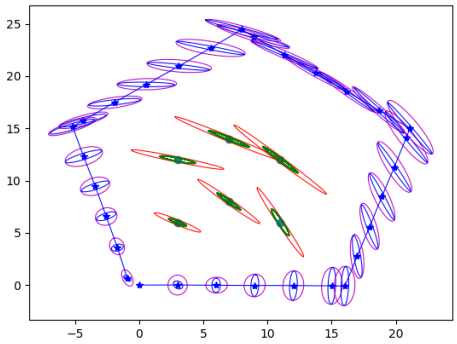
(b) Decreasing σ_α by Factor of 10

3.2.4 Increasing/Decreasing σ_β

Increasing/decreasing σ_β has the effect of increasing/decreasing the landmarks' uncertainty in the direction perpendicular to the initial line-of-sight from the robot to the landmarks. This is seen by the elongation of the covariance ellipse in the landmarks' perpendicular direction from the initial robot orientation pose. This is because σ_β determines the uncertainty in the bearing angle. A large σ_β value causes the uncertainty to "fan out". This is manifested at landmark initialization by a large uncertainty in the direction perpendicular to the initial line-of-sight from the robot to the landmarks. The converse is true when the σ_β value is decreased by a factor of 10. The robot's pose uncertainty also increases/decreases with increasing/decreasing σ_β since they are correlated. Higher/lower uncertainty in the landmark's location couples to the robot pose's uncertainty as it is less/more confident from the ray casting measurements.



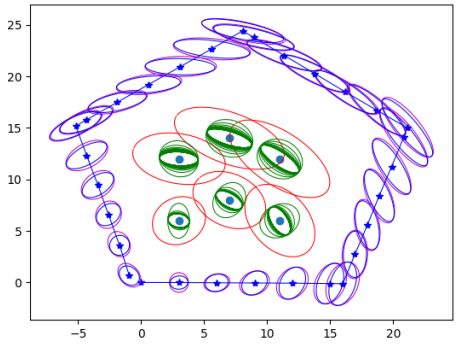
(a) Increasing σ_β by Factor of 10



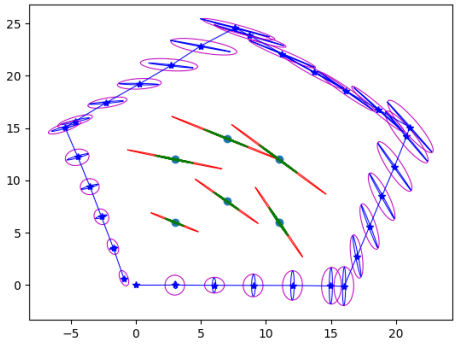
(b) Decreasing σ_β by Factor of 10

3.2.5 Increasing/Decreasing σ_r

Increasing/decreasing σ_r has the effect of increasing/decreasing the landmarks' uncertainty in the direction parallel to the initial line-of-sight from the robot to the landmarks. This is seen by the elongation of the covariance ellipse in the landmarks' parallel direction from the initial robot orientation pose. This is because σ_r determines the uncertainty in the range. A large σ_r value causes the distance to the landmark to be highly uncertain. This is manifested at landmark initialization by a large uncertainty in the direction parallel to the initial line-of-sight from the robot to the landmarks. The converse is true when the σ_r value is decreased by a factor of 10. The robot's pose uncertainty also increases/decreases with increasing/decreasing σ_r since they are correlated. Higher/lower uncertainty in the landmark's location couples to the robot pose's uncertainty as it is less/more confident from the ray casting measurements.



(a) Increasing σ_r by Factor of 10



(b) Decreasing σ_r by Factor of 10

3.3 Q3.

To achieve constant computational time per cycle when the total number of landmarks increase, the following changes to the framework can be made:

- Cap the number of landmarks and remove/store landmarks that have not been measured for extended periods of time.
- Strictly assume the landmarks are uncorrelated and only keep the diagonal sub-blocks of the landmark covariance matrix so only small matrices need to be inverted.
- Keep separate landmark maps for different regions visited. Only use the landmark map that is relevant to the current region.

- Determine the most informative landmarks (e.g. using principal component decomposition) and localize using this subset of landmarks.
- Use graph techniques such as GraphSLAM and maintain a factor graph instead of a dense matrix.
- Use a hybrid approach such as FastSLAM where the robot pose is estimated using a particle filter and the landmarks using a bank of EKFs.

4 Code Submission

The code is submitted to Gradescope under `ekf_slam.py`.

5 References

[1] Fausto, Julia. *Deriving Trigonometry Identities*. Accessed: February 15, 2025. 2024. URL: https://medium.com/maths-dover/deriving-trigonometry-identities-1407647076e2.

[2] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

[3] Wikipedia contributors. *Extended Kalman Filter – Non-additive Noise Formulation and Equations*. Accessed: February 15, 2025. 2024. URL: https://en.wikipedia.org/wiki/Extended_Kalman_filter#Non-additive_noise_formulation_and_equations.