Homework 2: SLAM using Extended Kalman Filter

16-833: Robot Localization and Mapping

Austin Windham

Theory

1. $$p_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}$$

control inputs: d_t & α_t

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

2.

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

$$G_t = \text{Jacobian of } p_{t+1} = \begin{bmatrix} 1 & 0 & -d_t \sin\theta_t \\ 0 & 1 & d_t \cos\theta_t \\ 0 & 0 & 1 \end{bmatrix}$$

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

$$\& \ R_t = H_t R H_t^T$$

$$\Sigma_t = N(0, \Sigma_t)$$

$$\Rightarrow \underline{\Sigma_{t+1} = G_t N(0, \Sigma_t) G_t^T + H_t R H_t^T}$$

3.

Given φ_t, β, r, n_β, n_r & $\varphi_t = [x_t, y_t, \theta_t]^T$

$$l_x = x_t + (r + n_r)\cos[\theta_t + \beta + n_\beta]$$

$$l_y = y_t + (r + n_r)\sin[\theta_t + \beta + n_\beta]$$

, by including r & B with their noises.

4. Find r and β

$$[l_x - x]^2 + (l_y - y)^2 = (r + n_r)^2$$, from dist. btw. landmark & robot

$$\Rightarrow r + n_r = \sqrt{(l_x - x)^2 + (l_y - y)^2}$$

& $l_x - x = \delta x$, $l_y - y = \delta y$

$$\Rightarrow r = \sqrt{\delta x^2 + \delta y^2} - n_r$$

$$\frac{\delta y}{\delta x} = \tan(\theta_t + \beta + n_\beta)$$

$$\Rightarrow \quad \beta = \tan^{-1}\left(\frac{\delta y}{\delta x}\right) - \theta_t - n_\beta$$

$$\beta = \text{wrap2pi}\left[\tan^{-1}\left(\frac{\delta y}{\delta x}\right) - \theta_t - n_\beta\right]$$

5. Find H_φ

$$H_\varphi = \begin{bmatrix} \frac{\partial B}{\partial x} & \frac{\partial B}{\partial y} & \frac{\partial B}{\partial \theta} \\ \frac{\partial r}{\partial x} & \frac{\partial r}{\partial y} & \frac{dr}{\partial \theta} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{1+\frac{\partial y^2}{\partial x^2}} \cdot \partial y \cdot x + \frac{x}{\partial x^2} & \frac{1}{1+\frac{\partial y^2}{\partial x^2}} \cdot \frac{-1}{\partial x} & -1 \\ \frac{-2\,\partial x}{2\sqrt{\partial x^2+\partial y^2}} & \frac{-2\,\partial y}{2\sqrt{\partial x^2+\partial y^2}} & 0 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\delta y}{\delta x^2+\delta y^2} & -\frac{\delta x}{\delta x^2+\delta y^2} & -1 \\ \frac{-\delta x}{\sqrt{\delta x^2+\delta y^2}} & -\frac{\delta y}{\sqrt{\delta x^2+\delta y^2}} & 0 \end{bmatrix}$$

6.

Find H_ℓ

$$H_\ell = \begin{bmatrix} \dfrac{\partial \beta}{\partial \ell_x} & \dfrac{\partial \beta}{\partial \ell_y} \\ \dfrac{\partial r}{\partial \ell_x} & \dfrac{\partial r}{\partial \ell_y} \end{bmatrix}$$

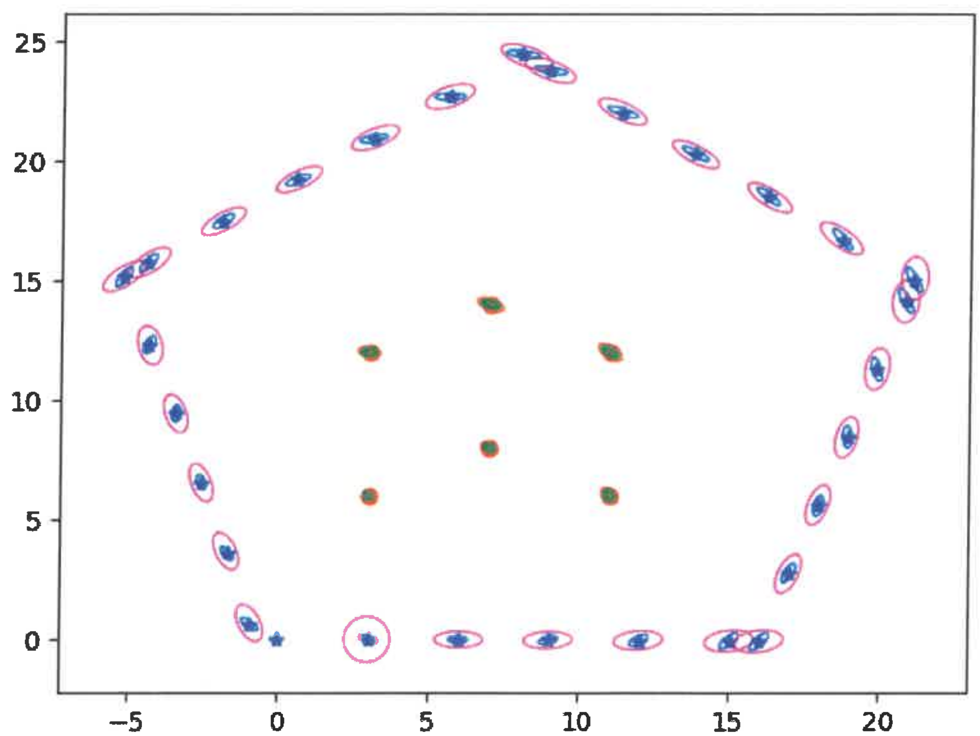
$$= \begin{bmatrix} \dfrac{-\delta y}{\delta x^2 + \delta y^2} & \dfrac{\delta x}{\delta x^2 + \delta y^2} \\ \dfrac{\delta x}{\sqrt{\delta x^2 + \delta y^2}} & \dfrac{\delta y}{\sqrt{\delta x^2 + \delta y^2}} \end{bmatrix}$$

We do not need to compute the Jacobian w.r.t. other landmarks because we are making the assumption that the measurements of landmarks to the robot's location are independent of one another.

Implementation:

1. Based on the data.txt file, there are 6 fixed landmarks being observed over the entire sequence.

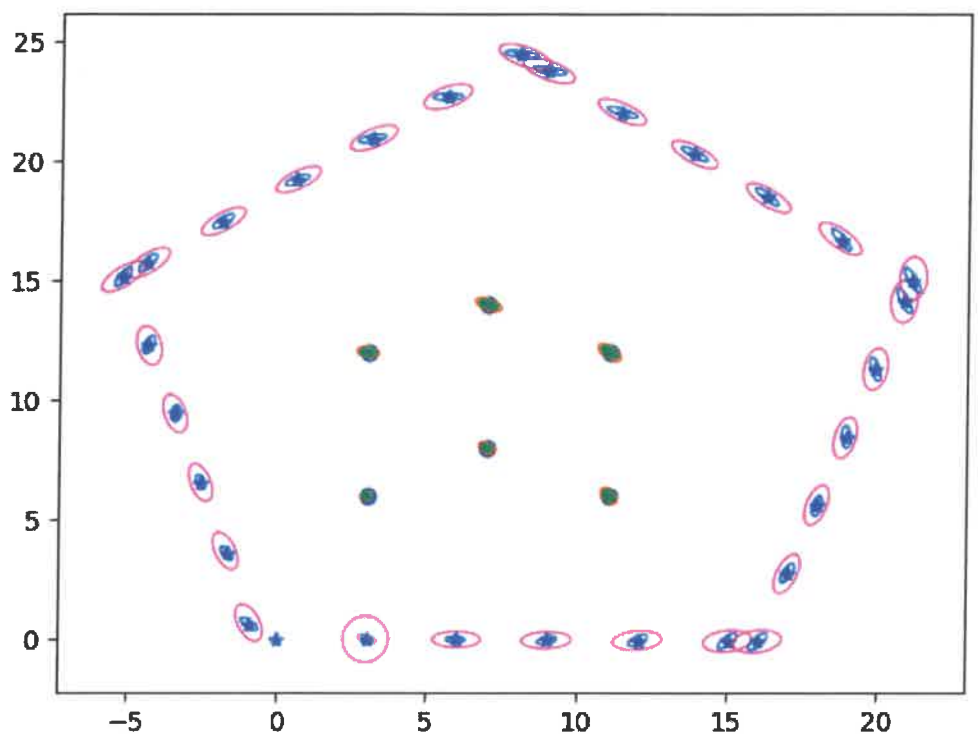
2. A visualization of the EKF simulation after all steps are finished is shown below. The original parameters were used in the visualization below.



3. The size of the ellipse around the robot's position decreases over time because the uncertainty is decreasing. The sizes of the ellipses around the landmarks also decrease over time, which also indicates that the uncertainties of the landmark positions are decreasing. So, the decreasing of size over time in these ellipses shows that the EKF-SLAM method is working to decrease the uncertainty of the position of the robot and landmarks over time.

EKF-SLAM reduces uncertainty by combining two key steps: the prediction and update steps. In each iteration, the filter first predicts the robot's new position and the landmark locations, increasing uncertainty. Then, it updates these estimates using sensor measurements, which reduces the uncertainty. This process allows the estimates to become more accurate over time, as reflected in the shrinking uncertainty ellipses.

4. In the plot below, the ground truth of the landmarks ae the six solid blue circles in the middle, and you can see that their centers lie inside the smallest ellipses. This indicates that the EKF-SLAM method has successfully reduced uncertainty over time. This also means that the true positions fall within the one-sigma distance of the estimated positions, showing that the uncertainty estimates are well-calibrated. The SLAM process effectively captures the true landmark positions with high confidence.



The Euclidean and Mahalanobis distances can be seen in the table below for the six landmarks. As you can see, these numbers are very small. The small Euclidean and Mahalanobis distances indicate that the estimated positions of the landmarks are very close to their true positions. The Euclidean distance shows high accuracy in the landmark location estimation, while the small Mahalanobis distance suggests that the estimates lie well within the expected uncertainty bounds. Overall, these results confirm that the EKF-SLAM algorithm has performed effectively in estimating both the positions and their associated uncertainties.

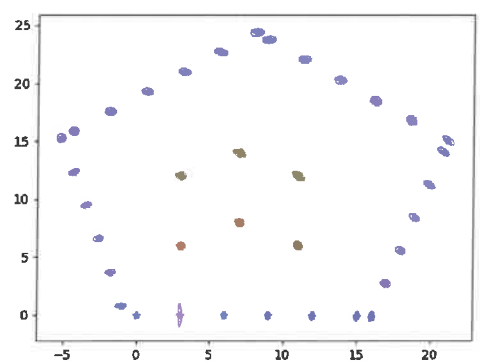
Landmark	Euclidean	Mahalanobis
1	0.002191704660480309	0.05750118627503789
2	0.004172732507936266	0.07660115444654661
3	0.002523110629758113	0.057850242168509244
4	0.0027936346213239768	0.07264924497304755
5	0.0019271182372538453	0.03284979069564497
6	0.003997375517185797	0.10767716415912029

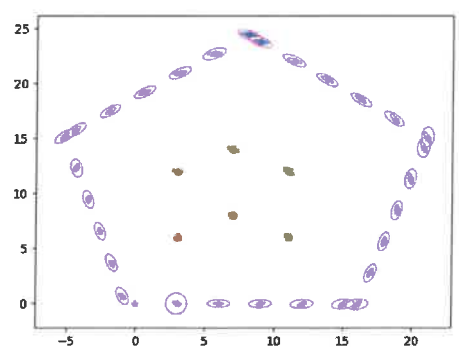
Discussion

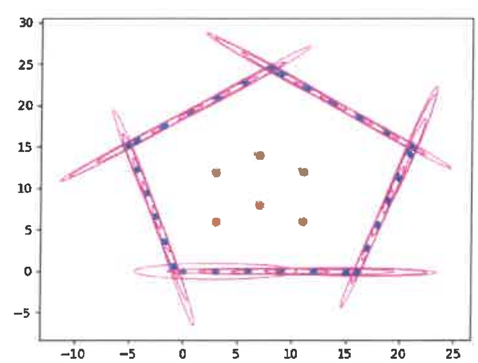
1. The reason why the zero terms in the initial landmark covariance matrix become non-zero in the final state covariance matrix is because during the update step of EKF-SLAM, the robot observes the landmarks and incorporates sensor measurements into the state estimation. These observations introduce uncertainty, as they are inherently noisy, and this uncertainty is propagated into both the robot's position and the landmarks' positions. As a result, the covariance matrix is updated to reflect these new and uncertain estimates, which create non-zero values representing the uncertainty relationships between the state elements.

The assumption made was that the landmarks are independent of one another. This is not necessarily correct because the landmarks are in fixed locations compared to one another, so a change in belief in one landmark should correspond to a change in belief of the location of the other landmarks.

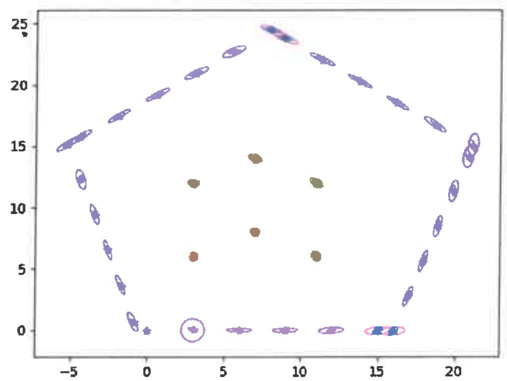
2. The visualizations below for sigma_x values of 0.025, 0.25, and 2.50 are below in that corresponding order. From the plots below, increasing sigma_x increases the uncertainty of the robot in its forward (x) direction while decreasing the value does the opposite.

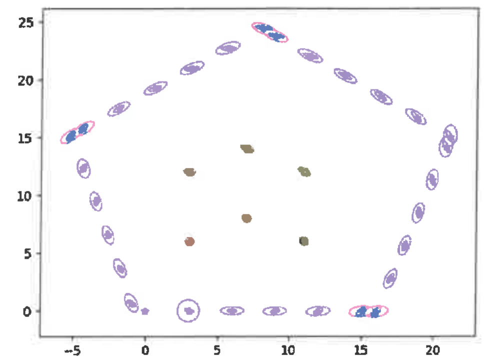


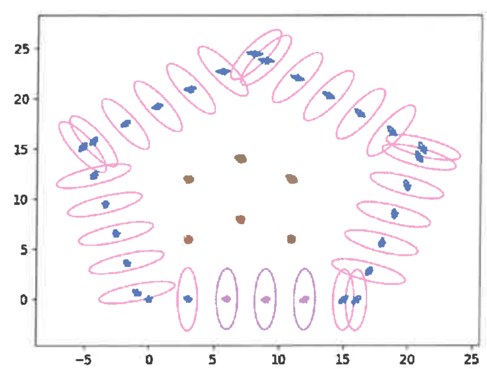




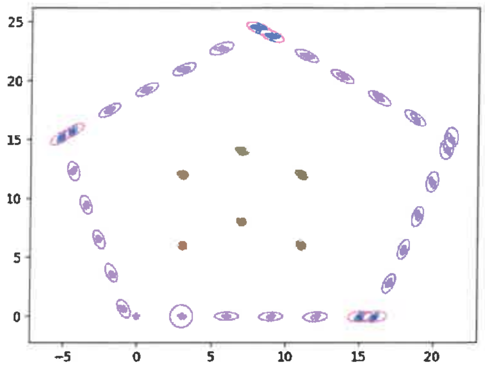
The visualizations for the varied sigma_y values of 0.01, 0.1, and 1.0 are below in that corresponding order. Like the results of the previous experiment, increasing sigma_y increases the uncertainty in the robot's lateral (y) direction, and decreasing sigma_y does the opposite.

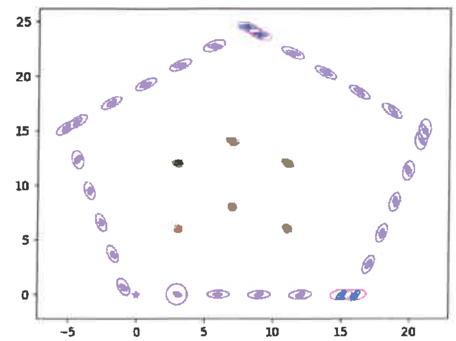


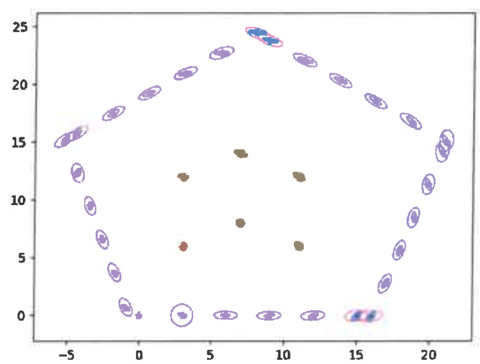




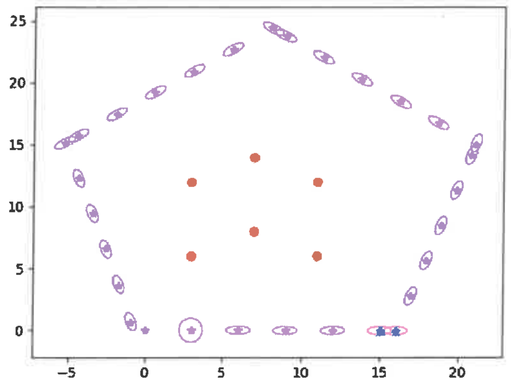
The visualizations for the varied sigma_alpha values of 0.01, 0.1, and 1.0 are below in that corresponding order. This value represents the standard deviation of the robot's orientation. It is difficult to see in these images, but when this value is increased, the uncertainty of the robot's orientation increases, and this can be seen particularly when the robot must turn, so you can see that there is more uncertainty at the corners of the pentagon.

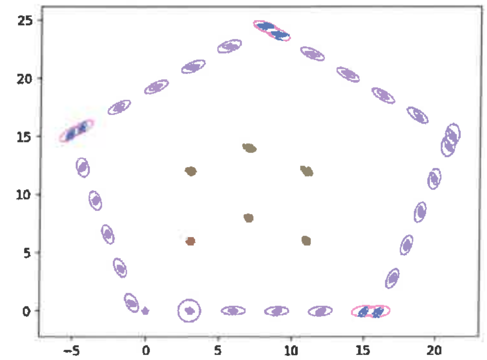


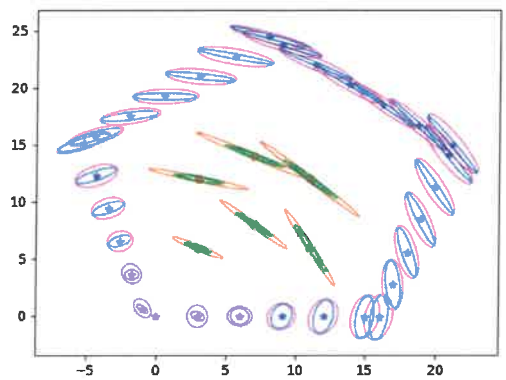




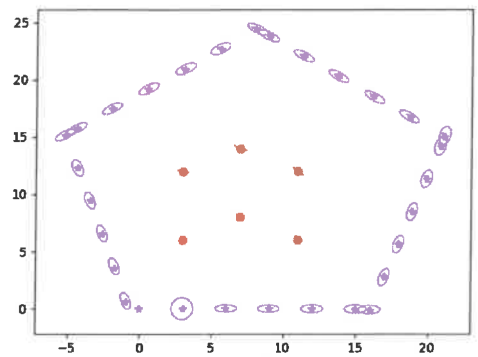
The visualizations for the varied sigma_beta values of 0.001, 0.01, and 0.1 are below in that corresponding order. Sigma_beta affects the uncertainty in the angular measurements of the landmark. When this is increased, the uncertainty of the landmarks clearly increases in the landmarks below. This increased uncertainty in the landmarks also causes an increased uncertainty in the robot's position.

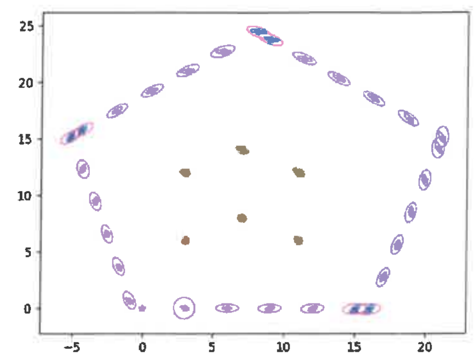


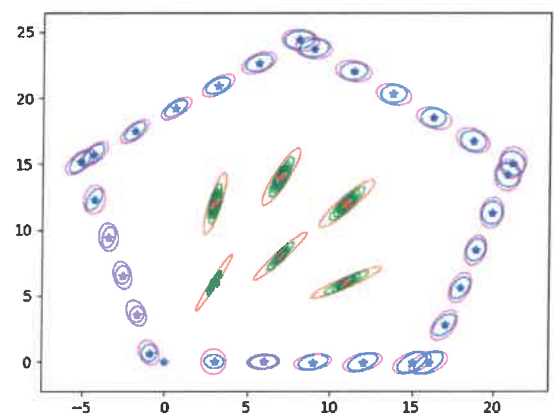




The visualizations for the varied sigma_r values of 0.008, 0.08, and 0.8 are below in that corresponding order. Sigma_r affects the uncertainty in the measurements of the distance from the landmarks and the robot. As you can see in the graphs, as this value is increased, the uncertainty in the landmark positions increases, which in turn causes uncertainty in the robot's position to increase.







3. There are several methods that we can use to make the EKF-SLAM framework achieve constant computation time or speed up the process. One thing we can do is called submap SLAM. This allows us to break up the area into smaller regions and then only focus on the landmarks in the region we are in. Another thing that can be done is that we can delete landmarks that we do not find useful because the robot will not revisit them again or because they offer redundant information. There are some heuristics for this deletion process. We can also only sample a set number of landmarks for each iteration. This may cause some losses in accuracy, but it will help with computation.