16833: Robot Localization and Mapping

HW2: SLAM using Extended Kalman Filter (EKF-SLAM)

Andrew ID: hdhruva

Discussed with anehete about landmark initialization.

1.

1. With no noise or error,

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

2. $e_x \sim \mathcal{N}(0, \sigma_x^2)$, $e_y \sim \mathcal{N}(0, \sigma_y^2)$, $e_\alpha \sim \mathcal{N}(0, \sigma_\alpha^2)$

P_t : Uncertainty of robots pose at time t $\sim \mathcal{N}(0, \Sigma_t)$

P_{t+1} : Uncertainty of robots pose at time $t+1$

We have, $$P_{t+1} = G_{t+1} P_t G_{t+1}^T + L_{t+1} R_t L_{t+1}^T$$

and with error, $$p_{t+1} = \begin{bmatrix} x_t + d_t\cos\theta_t + e_x\cos\theta_t - e_y\sin\theta_t \\ y_t + d_t\sin\theta_t + e_x\sin\theta_t + e_y\cos\theta_t \\ \theta_t + \alpha_t + e_\alpha \end{bmatrix}$$

$$G_{t+1} = \frac{\partial p_{t+1}(p_t, u, 0)}{\partial p_t} = \begin{bmatrix} 1 & 0 & -d_t\sin\theta_t \\ 0 & 1 & d_t\cos\theta_t \\ 0 & 0 & 1 \end{bmatrix}$$

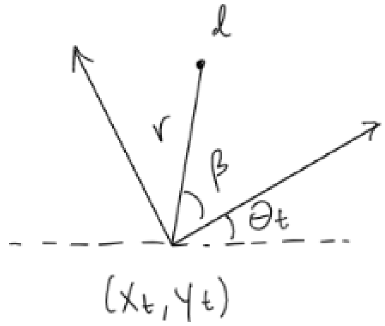
$$L_{t+1} = \frac{\partial p_{t+1}(p_t, u, 0)}{\partial e} = \begin{bmatrix} \cos\theta_t & -\sin\theta_t & 0 \\ \sin\theta_t & \cos\theta_t & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

Therefore,

$$P_{t+1} = \begin{bmatrix} 1 & 0 & -d_t\sin\theta_t \\ 0 & 1 & d_t\cos\theta_t \\ 0 & 0 & 1 \end{bmatrix} \mathcal{N}(0, \Sigma_t) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -d_t\sin\theta_t & d_t\cos\theta_t & 1 \end{bmatrix} + \begin{bmatrix} \cos\theta_t & -\sin\theta_t & 0 \\ \sin\theta_t & \cos\theta_t & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix} \begin{bmatrix} \cos\theta_t & \sin\theta_t & 0 \\ -\sin\theta_t & \cos\theta_t & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3.



$$\eta_\beta \sim \mathcal{N}(0, \sigma_\beta^2)$$
$$\eta_r \sim \mathcal{N}(0, \sigma_r^2)$$

$$l_x = X_t + r\cos(\theta_t + \beta + \eta_\beta)$$
$$l_y = Y_t + r\cos(\theta_t + \beta + \eta_\beta)$$

4. $m_x = l_x - x_t = r\cos(\theta_t + \beta + \eta_\beta)$

$m_y = l_y - y_t = r\sin(\theta_t + \beta + \eta_\beta)$

$$\theta_t + \beta + \eta_\beta = a\tan 2\left(\frac{m_y}{m_x}\right)$$

$$\beta = a\tan 2\left(\frac{m_y}{m_x}\right) - \eta_\beta - \theta_t$$

$$\boxed{\begin{aligned} \beta &= \mathrm{warp2pi}\left(\mathrm{np.arctan2}(m_y, m_x) - \theta_t\right) \\ r &= \sqrt{m_x^2 + m_y^2} \end{aligned}}$$

$$h = \begin{bmatrix} \beta \\ r \end{bmatrix}$$

5. $$H_p = \begin{bmatrix} \frac{\partial \beta}{\partial x_t} & \frac{\partial \beta}{\partial y_t} & \frac{\partial \beta}{\partial \theta_t} \\ \frac{\partial r}{\partial x_t} & \frac{\partial r}{\partial y_t} & \frac{\partial r}{\partial \theta_t} \end{bmatrix} = \begin{bmatrix} \frac{m_y}{r^2} & -\frac{m_x}{r^2} & -1 \\ \frac{m_x}{r^3} & \frac{m_y}{r^3} & 0 \end{bmatrix}$$

$$= \frac{1}{r^2} \begin{bmatrix} m_y & -m_x & -r^2 \\ \frac{m_x}{r} & \frac{m_y}{r} & 0 \end{bmatrix}$$

6. $$H_\ell = \begin{bmatrix} \frac{\partial \beta}{\partial \ell_x} & \frac{\partial \beta}{\partial \ell_y} \\ \frac{\partial r}{\partial \ell_x} & \frac{\partial r}{\partial \ell_y} \end{bmatrix} = \begin{bmatrix} -\frac{m_y}{r^2} & \frac{m_x}{r^2} \\ -\frac{m_x}{r^3} & -\frac{m_y}{r^3} \end{bmatrix} = \frac{1}{r^2} \begin{bmatrix} -m_y & m_x \\ -\frac{m_x}{r} & -\frac{m_y}{r} \end{bmatrix}$$

2.

2.1.

Based on the data format, there are two measurements per landmark, being bearing and range.

There are **6** fixed number of landmarks being observed over the entire sequence.

2.2.

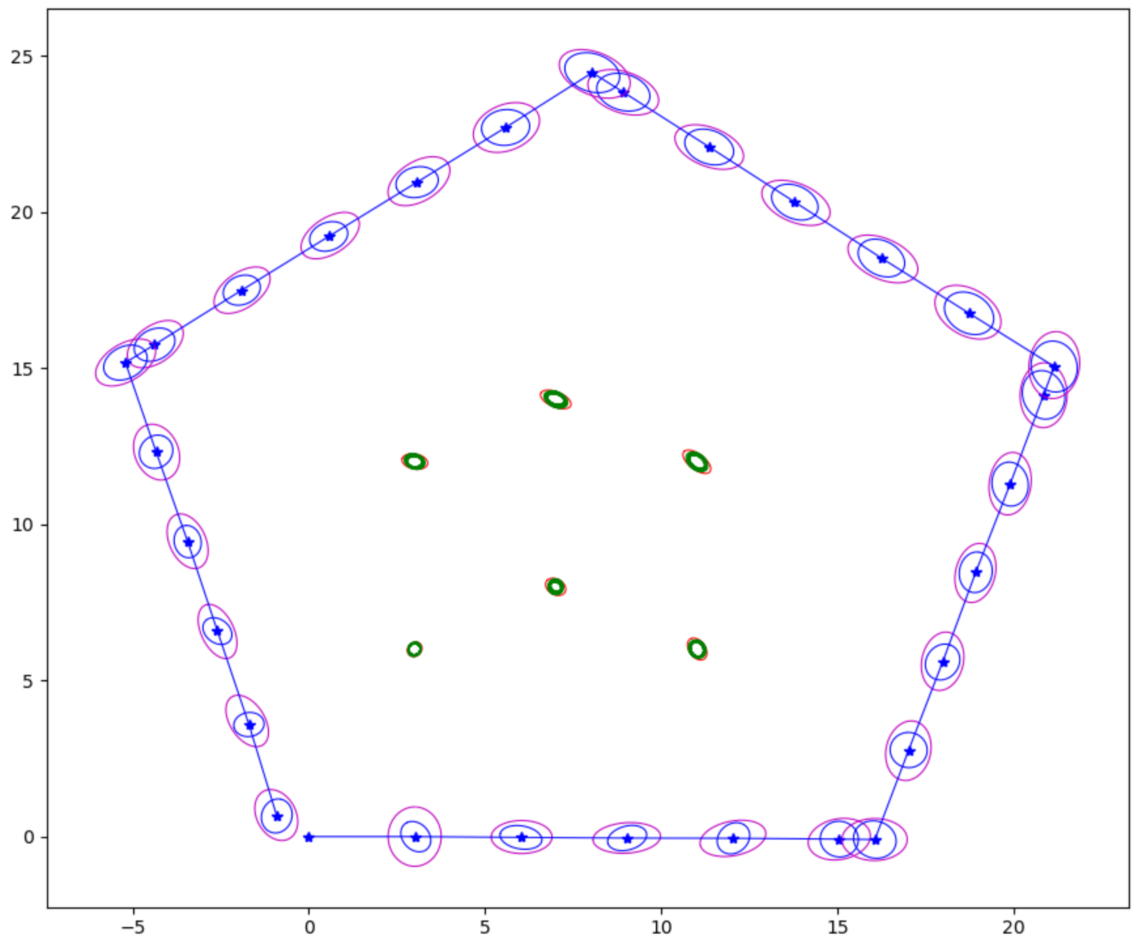


Figure 1: Visualization with default parameters

Visualization shown in Figure 1 once all steps are finished. The landmark covariances are very small therefore it is difficult to see the ellipses. Zoomed in version is shown in Figure 2.

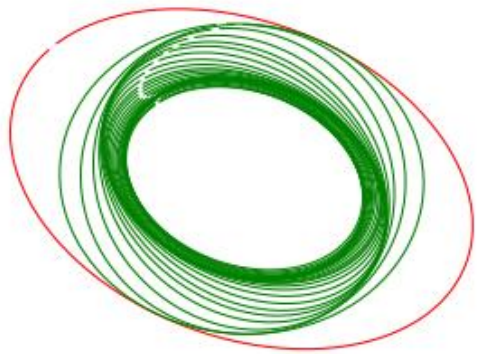


Figure 2: Zoomed in Landmark Covariance Ellipses

2.3.

As we see in the visualization, the predicted covariance of the robot poses, and the landmarks are higher than those after the update step. This reduction occurs due to the Kalman gain, which relatively weights the measurement and the previous state estimate. Therefore, the EKF result is a sort of weighted average of the measurement and the previous state. We can also see that the landmarks further away from the robot are initialized with a higher covariance, and as the robot traverses through, it is able to predict and correct more accurately.

Let us say the uncertainty of our sensor is very low, in this case the Kalman gain would take a higher value, which would then let the sensor measurement dominate, and therefore the low uncertainty sensor would tell us where we are. Say we know that our previous state uncertainty is very low, then a lower Kalman gain would weight towards the previous state estimate, ignoring most of the measurement.

Therefore, EKF-SLAM improves the estimation and trajectory by comparing uncertainty ellipses and weighting each input accordingly, based on the uncertainties.

2.4.

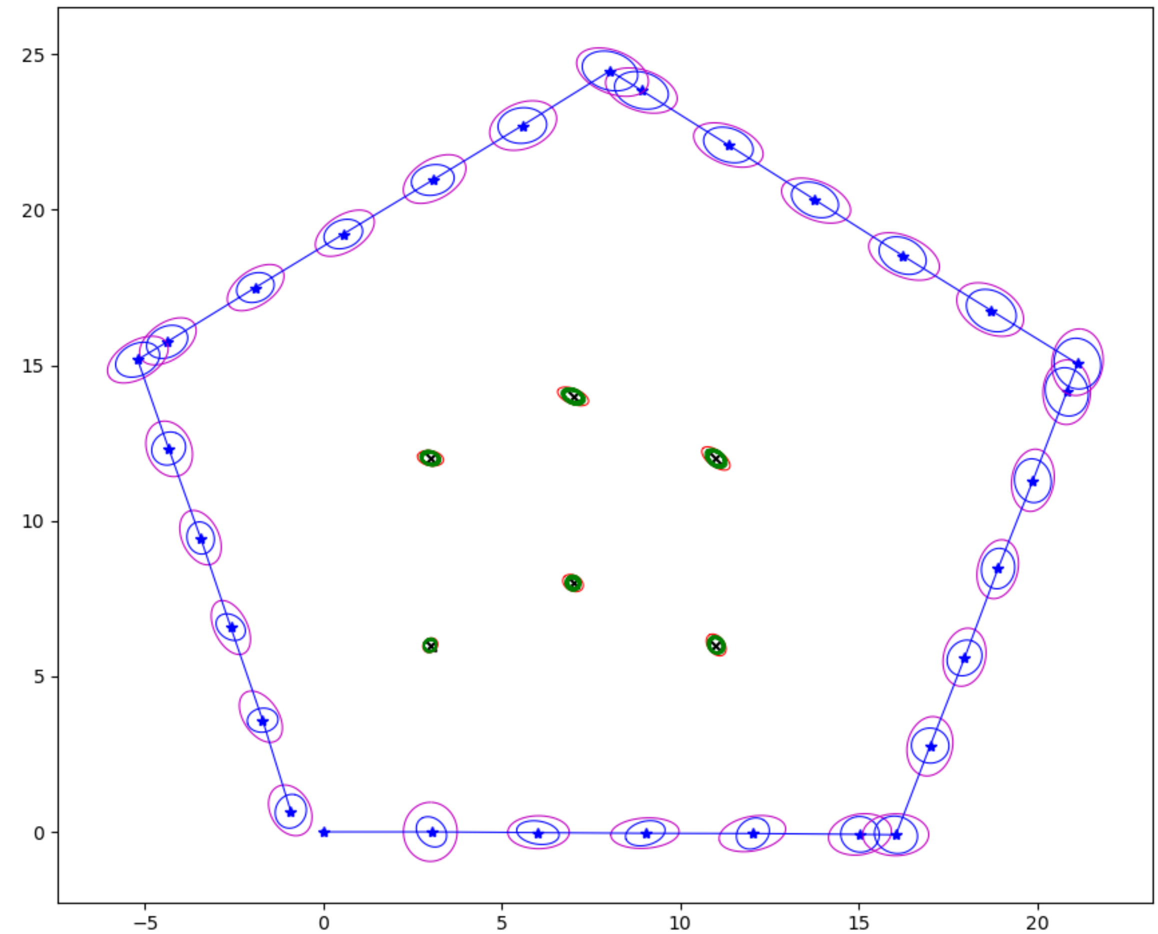


Figure 3: Visualization with ground truth and default parameters

The ground truths of the landmarks are plotted in Figure 3 as **X**, and they lie inside the smallest corresponding ellipse. This is also shown zoomed in, in Figure 4. This means that the true location of the landmarks is very close to the estimated location by the EKF.

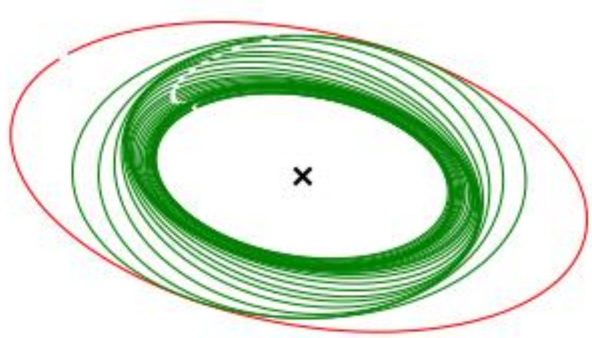


Figure 4: Zoomed in Landmark covariance Ellipses

Table1: Landmark Euclidean and Mahalanobis distances

Landmark	Euclidean distance	Mahalanobis distance
1	0.00526844	0.00028343
2	0.00626463	0.00034913
3	0.00076271	0.00003893
4	0.00536144	0.00028728
5	0.00884154	0.00052986
6	0.00764498	0.00052868

The Euclidean distance computes the geometric distance, so in our case in 2D, we are very close to the ground truth.

The Mahalanobis distance indicates the distance between a point and the distribution. In our case the distances are very low, which means the mean of the resultant distribution is very close to the true location of the landmark.

3.

3.1.

The zero terms in the initial landmark covariance matrix become non-zero in the final state covariance matrix because the landmarks are dependent on each other, or are correlated, as the robot state updates. The landmarks are also dependent on the robot state, as the landmark covariance matrix was initialized accounting for both the initial pose of the robot and the initial measurement covariance. As the robot updates it's state and traverses the path, we get better estimates of the landmarks that are closer to the robot's field of view, as the variance in bearing and range would be lower, when the robot is closer.

When setting the initial value for cross covariance matrix P, we assume the cross correlation between the landmarks and the robot state is zero as we set all those values to zero. This assumption is not necessarily correct.

3.2.

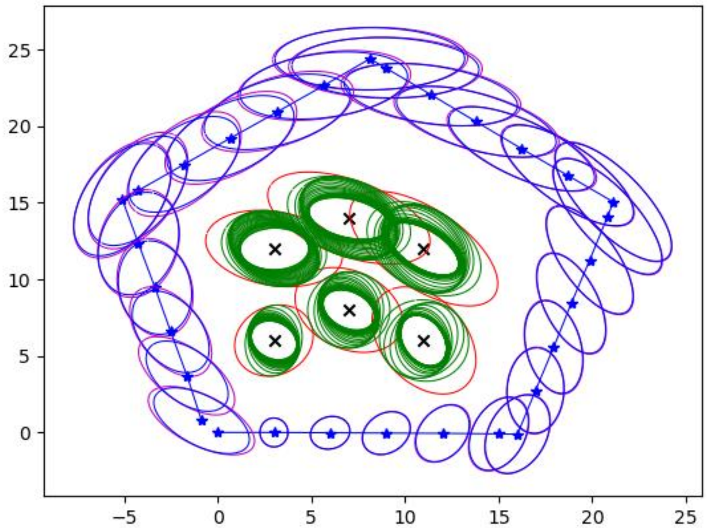


Figure 5: $\sigma_\alpha, \sigma_\beta, \sigma_r \rightarrow 10 * X$

Here we can see the increased covariance in both the robot's pose and the landmarks, the robot's pose covariance increasing in the angular direction.

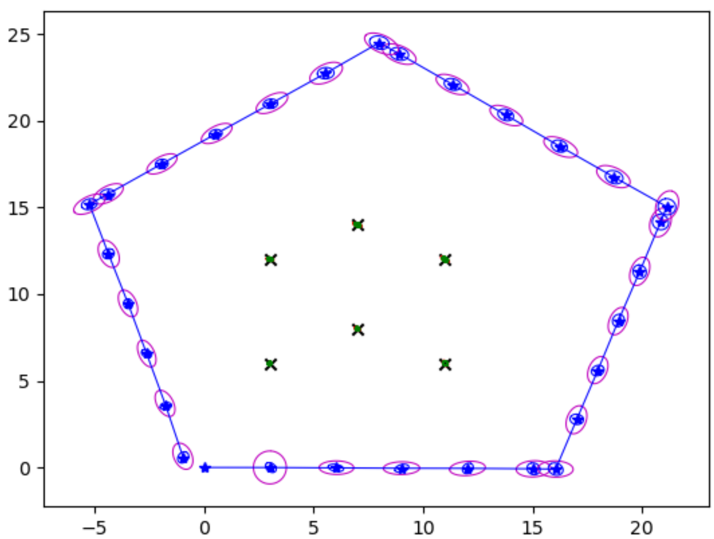


Figure 6: $\sigma_\alpha, \sigma_\beta, \sigma_r \rightarrow 0.5 * X$

Reducing these parameters shows significant decrease in the size of the covariance ellipses, for the robot's pose as well as the landmarks.

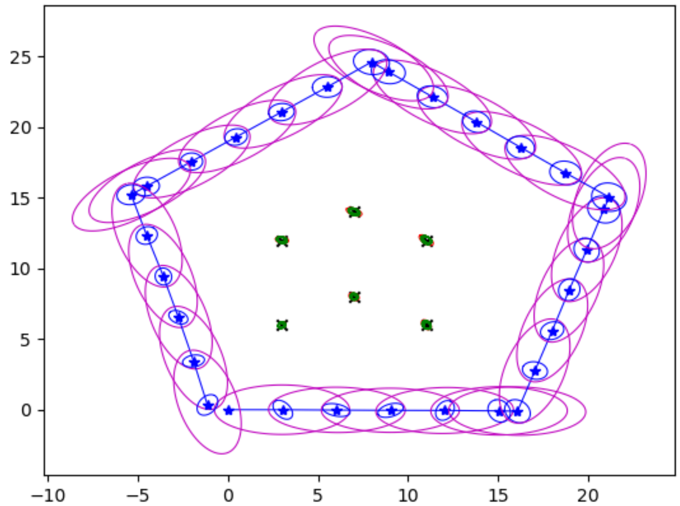


Figure 7: $\sigma_x, \sigma_y \rightarrow 5 * X$

We can observe the increase variance in the robot's pose, as the corresponding x and y $\sigma's$ are increased. Here the landmark's variance remains small relatively.

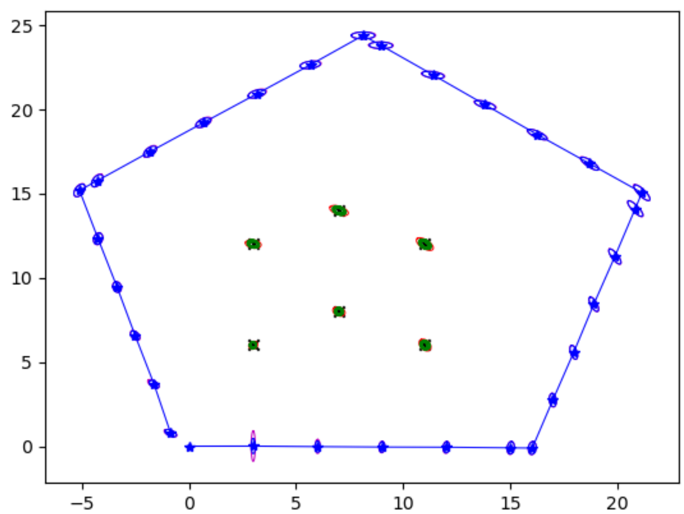


Figure 8: $\sigma_x, \sigma_y \rightarrow 0.1 * X$

We can observe a drastic decreas in variance in the robot's pose, as the corresponding x and y $\sigma's$ are decreased. Here the landmark's variance remains relatively the same, as those $\sigma's$ are unchanged.

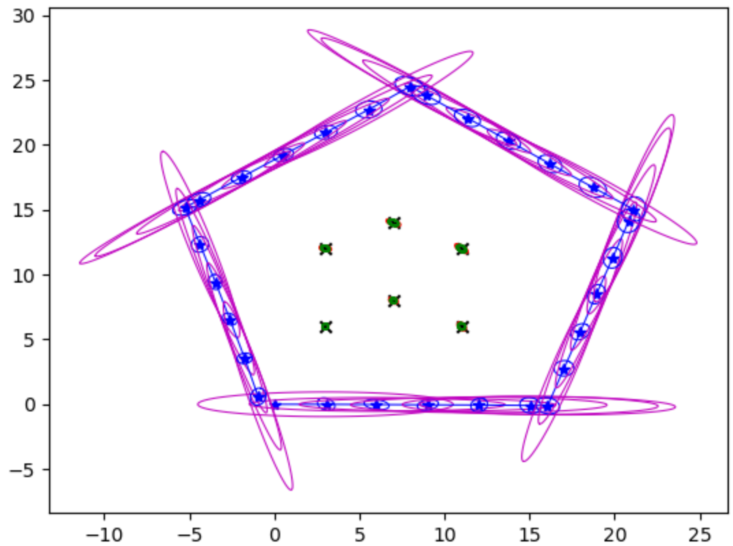


Figure 9: $\sigma_x \rightarrow 10 * X$

Increase in only σ_x shows an increase in the covariance in the x direction as the robot updates it's pose.

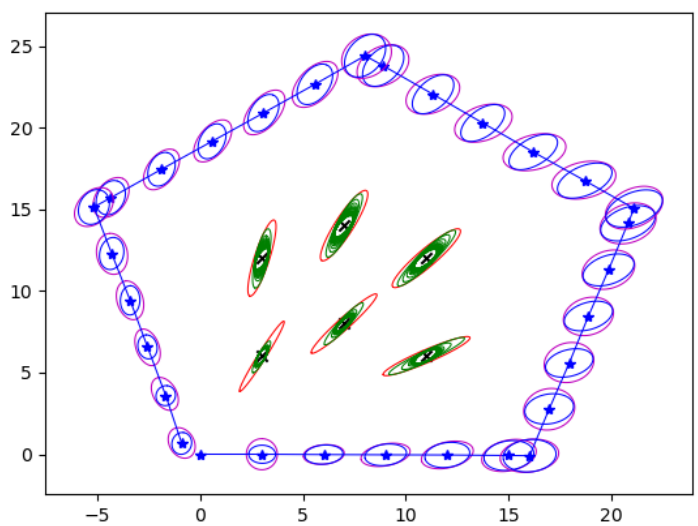


Figure 10: $\sigma_r \rightarrow 10 * X$

Increase in only σ_r shows an increase in the covariance in the direction of r, as the robot faces it when initialized. Therefore, the landmark covariance ellipses are elongated in that direction. The further landmarks also show higher elongation in the angular β direction.

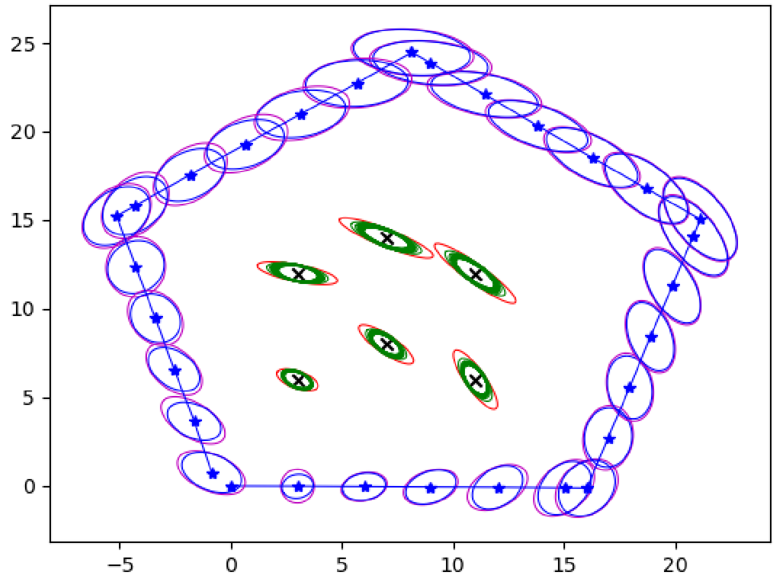


Figure 11: $\sigma_\beta, \sigma_r \to 5 * X, 2 * X$

Increase in only σ_β, with a slight increase in σ_r shows an increase in the covariance in the angular direction of β, as the robot faces it when initialized. Therefore, the landmark covariance ellipses are elongated in the angular direction. The further landmarks from the initialized position show a higher elongation in the covariance ellipses.

3.3.

Potential changes that could be made to account for increasing landmarks

- Update landmarks based on proximity. This could be achieved by optimizing control inputs, for the robot to properly explore a given area, after which the landmarks in the area would have a reasonable estimate. Then the control inputs could increase strategically, so that the robot moves away from the estimated landmarks, and if a new landmark is identified, a fairly estimated landmark will be replaced in the matrix based on some threshold. This would keep the number of landmarks in the "field of view" of the robot to a limited number but would require re-initialization of landmarks based on the current estimated robot pose. Loop closure techniques could be used to further estimate the forgotten landmarks. This would be slower, in terms of requiring redundant exploration by the robot.
- We could initialize a very large number of landmarks and fill in the sparse matrix based on the observed landmarks. Using sparse matrix techniques could help with the speed of computation. However, with an unbounded amount of landmarks, this approach could fail.
- Another potential approach would approximate several groups of landmarks by say a center landmark representing a cluster of landmarks that are estimated to be close. As the robot keeps seeing new landmarks, earlier mentioned approaches could be paired, to keep the number of landmarks observed to be low.