16-833 HW2:
SLAM using Extended Kalman Filter (EKF-SLAM)

Yash Jain (Andrew ID - yashjain)

1 Theory

1.1 Next Pose p_{t+1}

Given current pose $p_t = [x_t, y_t, \theta_t]^T$ and control input be $u_t = [d_t \cdot cos(\theta_t), d_t \cdot sin(\theta_t), \alpha_t]^T$. If there exists no noise, the next pose prediction is given as:

$$p_{t+1} = p_t + \begin{bmatrix} d_t \cdot cos(\theta_t) \\ d_t \cdot sin(\theta_t) \\ \alpha_t \end{bmatrix} \tag{1}$$

$$p_{t+1} = \begin{bmatrix} x_t + d_t \cdot cos(\theta_t) \\ y_t + d_t \cdot sin(\theta_t) \\ \theta_t + \alpha_t \end{bmatrix} \tag{2}$$

1.2 Next Pose p_{t+1} with uncertainty

Given the uncertainty in x, y and θ as $e_x \sim \mathcal{N}(0, \sigma_x^2)$, $e_y \sim \mathcal{N}(0, \sigma_y^2)$ and $e_\alpha \sim \mathcal{N}(0, \sigma_\alpha^2)$ respectively, the next pose p_{t+1} is given as:

$$p_{t+1} = \begin{bmatrix} x_t + (d_t + e_x) \cdot cos(\theta_t) - e_y \cdot sin(\theta_t) \\ y_t + (d_t + e_y) \cdot sin(\theta_t) + e_t \cdot cos(\theta_t) \\ \theta_t + \alpha_t + e_\alpha \end{bmatrix} \tag{3}$$

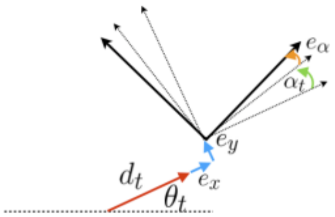


Figure 1: The robot frame of reference

The covariance estimate Σ_{t+1} can be calculated as:

$$\Sigma_{t+1} = G_t \cdot \Sigma_t \cdot G_t^T + L_t \cdot Q_t \cdot L_t^T \tag{4}$$

where G_t and L_t are the Jacobian matrices w.r.t the pose and the process errors and can be calculated as:

$$
\begin{aligned}
G_t &= \frac{\partial g}{\partial p}|_{p_t,u_t} && (5)\\
&= \begin{bmatrix} \frac{\partial g_1}{\partial x_t} & \frac{\partial g_1}{\partial y_t} & \frac{\partial g_1}{\partial \theta_t} \\ \frac{\partial g_2}{\partial x_t} & \frac{\partial g_2}{\partial y_t} & \frac{\partial g_2}{\partial \theta_t} \\ \frac{\partial g_3}{\partial x_t} & \frac{\partial g_3}{\partial y_t} & \frac{\partial g_3}{\partial \theta_t} \end{bmatrix} && (6)\\
&= \begin{bmatrix} 1 & 0 & -d_t \cdot sin(\theta_t) \\ 0 & 1 & d_t \cdot cos(\theta_t) \\ 0 & 0 & 1 \end{bmatrix} && (7)
\end{aligned}
$$

and

$$
\begin{aligned}
L_t &= \frac{\partial f}{\partial w}|_{p_t,u_t} && (8)\\
&= \begin{bmatrix} \frac{\partial g_1}{\partial e_x} & \frac{\partial g_1}{\partial e_y} & \frac{\partial g_1}{\partial e_\alpha} \\ \frac{\partial g_2}{\partial e_x} & \frac{\partial g_2}{\partial e_y} & \frac{\partial g_2}{\partial e_\alpha} \\ \frac{\partial g_3}{\partial e_x} & \frac{\partial g_3}{\partial e_y} & \frac{\partial g_3}{\partial e_\alpha} \end{bmatrix} && (9)\\
&= \begin{bmatrix} cos(\theta_t) & -sin(\theta_t) & 0 \\ sin(\theta_t) & cos(\theta_t) & 0 \\ 0 & 0 & 1 \end{bmatrix} && (10)
\end{aligned}
$$

and the covariance of process error Q_t is:

$$
Q_t = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\alpha^2 \end{bmatrix} \tag{11}
$$

1.3 Estimated Position of Landmark l_x, l_y

The landmark measurement noises are given as:

$$
\begin{aligned}
n_\beta &\sim \mathcal{N}(0, \sigma_\beta^2)\\
n_r &\sim \mathcal{N}(0, \sigma_r^2)
\end{aligned} \tag{12}
$$

The estimated position of landmark (l_x, l_y) is given as:

$$
\begin{bmatrix} l_x \\ l_y \end{bmatrix} = \begin{bmatrix} x_t + r_t \cdot cos(\theta_t + \beta_t) \\ y_t + r_t \cdot sin(\theta_t + \beta_t) \end{bmatrix} \tag{13}
$$

The uncertainty of landmark location estimation Σ_{l_t} w.r.t pose covariance and measurement covariance is calculated as:

$$
\Sigma_{l_{t+1}} = G_{l_t} \cdot \Sigma_t \cdot G_{l_t}^T + L_{l_t} \cdot R_t \cdot L_{l_t}^T \tag{14}
$$

where G_{l_t} and L_{l_t} are the Jacobian matrices w.r.t the pose and the measurement, and can be calculated as:

$$
\begin{aligned}
G_{l_t} &= \frac{\partial l}{\partial p}|_{p_t} && (15)\\
&= \begin{bmatrix} \frac{\partial l_1}{\partial x_t} & \frac{\partial l_1}{\partial y_t} & \frac{\partial l_1}{\partial \theta_t} \\ \frac{\partial l_2}{\partial x_t} & \frac{\partial l_2}{\partial y_t} & \frac{\partial l_2}{\partial \theta_t} \end{bmatrix} && (16)\\
&= \begin{bmatrix} 1 & 0 & -r_t \cdot sin(\theta_t + \beta_t) \\ 0 & 1 & r_t \cdot cos(\theta_t + \beta_t) \end{bmatrix} && (17)
\end{aligned}
$$

and

$$L_{l_t} = \frac{\partial l}{\partial z}|_{p_t} \tag{18}$$

$$= \begin{bmatrix} \frac{\partial l_1}{\partial \beta_t} & \frac{\partial l_1}{\partial r_t} \\ \frac{\partial l_2}{\partial \beta_t} & \frac{\partial l_2}{\partial r_t} \end{bmatrix} \tag{19}$$

$$= \begin{bmatrix} -r_t \cdot sin(\theta_t + \beta_t) & cos(\theta_t + \beta_t) \\ r_t \cdot cos(\theta_t + \beta_t) & sin(\theta_t + \beta_t) \end{bmatrix} \tag{20}$$

and the covariance of measurement noises R_t is given by:

$$R_t = \begin{bmatrix} \sigma_\beta^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix} \tag{21}$$

1.4 Measurement of Bearing and Range

Using the results in the section 1.3, we can calculate the Bearing (β) and Range (r) as follows:

$$\begin{bmatrix} \beta_t \\ r_t \end{bmatrix} = \begin{bmatrix} warp2pi(arctan2(l_y - y_t, l_x - x_t) - \theta_t) \\ \sqrt{(l_x - x_t)^2 + (l_y - y_t)^2} \end{bmatrix} \tag{22}$$

1.5 Measurement Jacobian H_p w.r.t Robot Pose

The measurement Jacobian H_p w.r.t the robot pose can be calculated as:

$$H_p == \frac{\partial h}{\partial p}|_{p_t} \tag{23}$$

$$= \begin{bmatrix} \frac{\partial h_1}{\partial x_t} & \frac{\partial h_1}{\partial y_t} & \frac{\partial h_1}{\partial \theta_t} \\ \frac{\partial h_2}{\partial x_t} & \frac{\partial h_2}{\partial y_t} & \frac{\partial h_2}{\partial \theta_t} \end{bmatrix} \tag{24}$$

$$= \begin{bmatrix} \frac{l_y - y_t}{(l_x - x_t)^2 + (l_y - y_t)^2} & \frac{-(l_x - x_t)}{(l_x - x_t)^2 + (l_y - y_t)^2} & -1 \\ \frac{-(l_x - x_t)}{\sqrt{(l_x - x_t)^2 + (l_y - y_t)^2}} & \frac{-(l_y - y_t)}{\sqrt{(l_x - x_t)^2 + (l_y - y_t)^2}} & 0 \end{bmatrix} \tag{25}$$

1.6 Measurement Jacobian H_l w.r.t Landmark

The measurement Jacobian H_p w.r.t landmark can be calculated as:

$$H_l == \frac{\partial h}{\partial l}|_{p_t} \tag{26}$$

$$= \begin{bmatrix} \frac{\partial h_1}{\partial l_x} & \frac{\partial h_1}{\partial l_y} \\ \frac{\partial h_2}{\partial l_x} & \frac{\partial h_2}{\partial l_y} \end{bmatrix} \tag{27}$$

$$= \begin{bmatrix} \frac{-(l_y - y_t)}{(l_x - x_t)^2 + (l_y - y_t)^2} & \frac{l_x - x_t}{(l_x - x_t)^2 + (l_y - y_t)^2} \\ \frac{l_x - x_t}{\sqrt{(l_x - x_t)^2 + (l_y - y_t)^2}} & \frac{l_y - y_t}{\sqrt{(l_x - x_t)^2 + (l_y - y_t)^2}} \end{bmatrix} \tag{28}$$

We do not calculate the Jacobian w.r.t the other landmarks based on the assumption that all landmarks are independent of each other.

2 Implementation and Evaluation

2.1 Number of Landmarks

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

2.2 Visualization

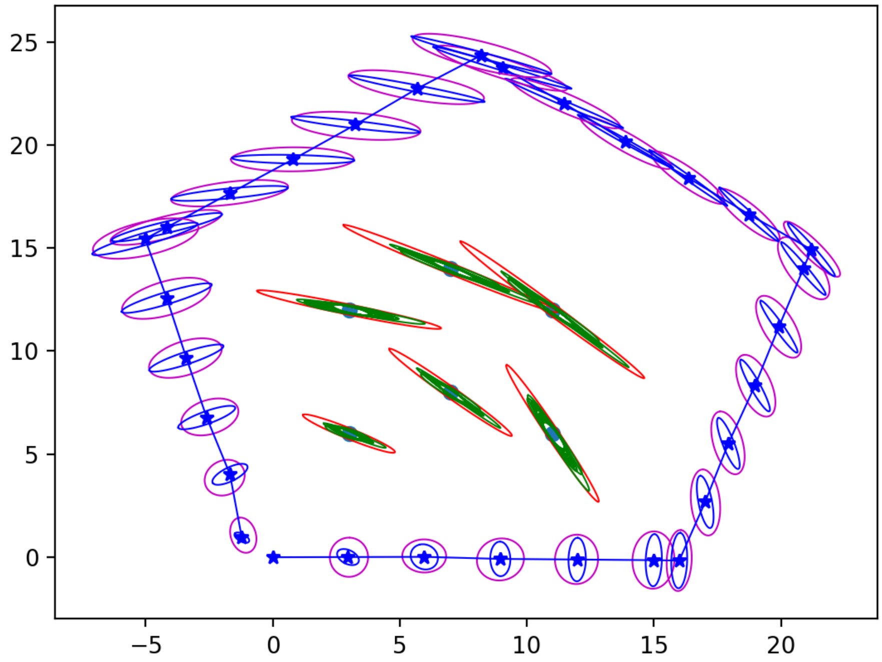


Figure 2: Visualization of EKF output after all steps.

2.3 EKF-SLAM Improvement

Through the update step in EKF-SLAM algorithm, we see an improvement in the estimation trajectory and map as the updated uncertainty ellipses are always smaller than the predicted uncertainty ellipses.

2.4 Evaluation of Performance

We can see in Figure 2, that the ground truth positions (represented by the green points) are inside the smallest corresponding ellipse, which means that the EKF was able to reduce the uncertainty in the estimate of the landmark position. The Euclidean distances and Mahalanobis distances w.r.t each landmarks are as follows:

Landmark	Euclidean	Mahalanobis
1	0.06210787	0.49023215
2	0.11307297	0.53430003
3	0.11695076	0.52277246
4	0.16491417	0.51687979
5	0.14822901	0.5735667
6	0.19670177	0.59483382

Since the distances are very small we can infer that the estimates are very close to the ground truth.

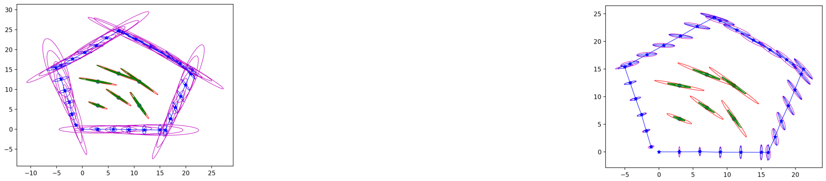
3 Discussion

3.1 Initial Covariance

The zero terms in the initial landmark covariance become non zero in the final state covariance matrix because the Kalman gain becomes non zero which in turn alters the non diagonal elements of the covariance matrix. Thus we can infer that the landmarks positions are correlated with each other and also the landmarks are correlated with the pose. The assumption made regarding the cross covariances is that there is no cross covariance, i.e., the variation in x direction does not effect y direction, the landmarks are independent of each other and the landmarks are independent of the pose. The updated non zero elements in the covariance matrix proves that these assumptions are incorrect.

3.2 Parameters

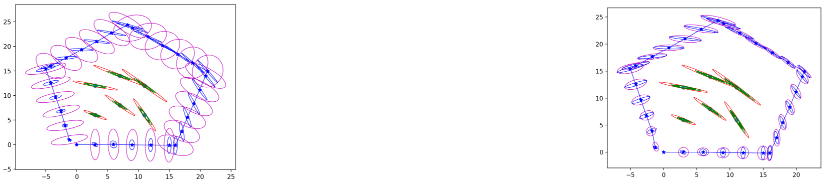
Each parameter was changed by a factor of 10 in both the positive and negative direction. The results obtained are represented in the figures below-



(a) $\sigma_x = 2.5$

(b) $\sigma_x = 0.025$

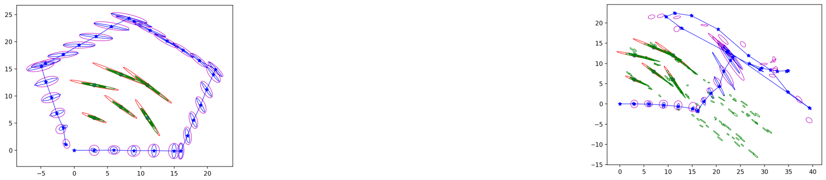
Figure 3: $\sigma_x variation$



(a) $\sigma_y = 1$

(b) $\sigma_y = 0.01$

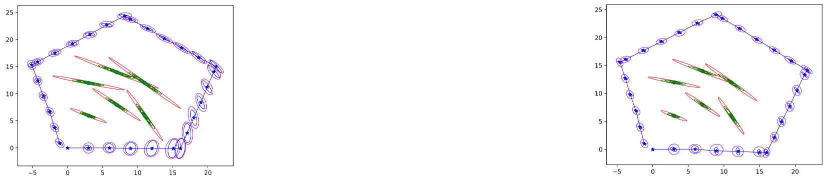
Figure 4: $\sigma_y variation$



(a) $\sigma_\alpha = 1$

(b) $\sigma_\alpha = 0.01$

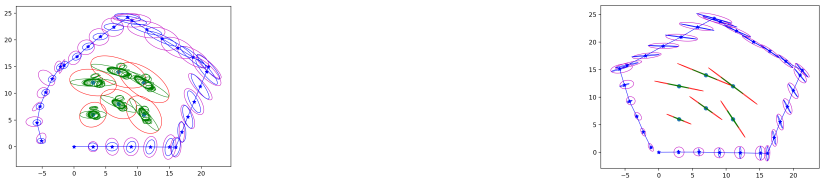
Figure 5: $\sigma_\alpha variation$



(a) $\sigma_\beta = 0.1$

(b) $\sigma_\beta = 0.001$

Figure 6: $\sigma_\beta variation$



(a) $\sigma_r = 0.8$

(b) $\sigma_r = 0.008$

Figure 7: $\sigma_r variation$

We can see a trend here. As the covariance increases, the uncertainty in that direction also increases along with the distance from the ground truth of landmarks. When the covariance decreases, the uncertainty in that direction also decreases along with the distance from the ground truth of landmarks except in the case of σ_α where the covariance becomes so small that the update does not account for errors and the EKF becomes unreliable.

3.3 Real World Landmarks

When the robot explores new environments, the list of landmarks can go unbounded but we know that there is correspondence between landmarks in the real world. Thus, when a landmark is no longer observed, we can store it in memory and remove it from the pose matrix to keep the size of the pose matrix fixed. This way we can achieve constant computation time in each cycle.