CSE-483: Mobile Robotics **Monsoon 2019**

Assignment - 4

Prepared by: AadilMehdi Sanchawala, Rahul Sajnani

In this report, we discuss our approach to doing the assignment questions. Instructions to run the assignment are present in the end of this report.

1 Localization using sensor fusion - (ekf.py)

1.1 Extended Kalman filter

Extended Kalman filter estimation algorithm is used for optimally estimating the position of the robot in a noisy environment using measurements observed over a trajectory. Extended Kalman filter is used mainly when the robot's position and sensor functions are non linear. The algorithm incorporates the sensor and control noise while predicting the position of the robot. This algorithm assumes Gaussian noise model.

1.2 Algorithm

1.2.1 Initialization

We initialize our position value with the first ground truth position. The covariance matrix initialized as diag(1,1,0.1).

1.2.2 Prediction

1. Position estimation

$$X_k = X_{k-1} + dt * \begin{bmatrix} cos\theta_{k-1} & 0 \\ sin\theta_{k-1} & 0 \\ 0 & 1 \end{bmatrix} * \left(\begin{bmatrix} v_k \\ w_k \end{bmatrix} + w_k \right) \tag{1}$$

2. Since our position function is non linear we find its jacobian as follows. Derivation of jacobian is provided at the end of the document.

$$G = \begin{bmatrix} 1 & 0 & -dt * v_k * sin\theta_{k-1} \\ 0 & 1 & dt * v_k * cos\theta_{k-1} \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

3. We estimate the covariance matrix

$$Cov_k = G * Cov_{k-1} * G^T \tag{3}$$

1.2.3 Correction

1. Calculate jacobian for observation.
   r_k^l = Range of landmark k and robot's laser rangefinder.

d = Distance between center of robot and laser rangefinder.
Q = covariance matrix of observation noise.
ϕ_k^l = Bearing to each landmark.
Angles are normalized to range -180 to 180 degrees Derivation of jacobian matrix is provided towards the end of the document.

$$H = \begin{bmatrix} \frac{\partial r_k^l}{\partial x_k} & \frac{\partial r_k^l}{\partial y_k} & \frac{\partial r_k^l}{\partial \theta_k} \\ \frac{\partial \phi_k^l}{\partial x_k} & \frac{\partial \phi_k^l}{\partial y_k} & \frac{\partial \phi_k^l}{\partial \theta_k} \\ \vdots & \vdots & \vdots \end{bmatrix} \tag{4}$$

Note: Above matrix is 2L x 3 dimensional matrix where L is the number of landmarks. Values for not visible landmarks are equated to 0.

2. Calculate Kalman gain K

$$K = Cov_k * H^T * (H * Cov_k * H^T + Q)^{-1} \tag{5}$$

3. Correcting mean position and covariance.
Ob = Calculated observation computed from position X_k

$$X_k = X_k + K * (Z - Ob) \tag{6}$$
$$Cov_k = (I - K * H) * Cov_k \tag{7}$$

Note: The observation jacobian and observation readings that are not visible from position X_k are equated to 0.

2 Outputs

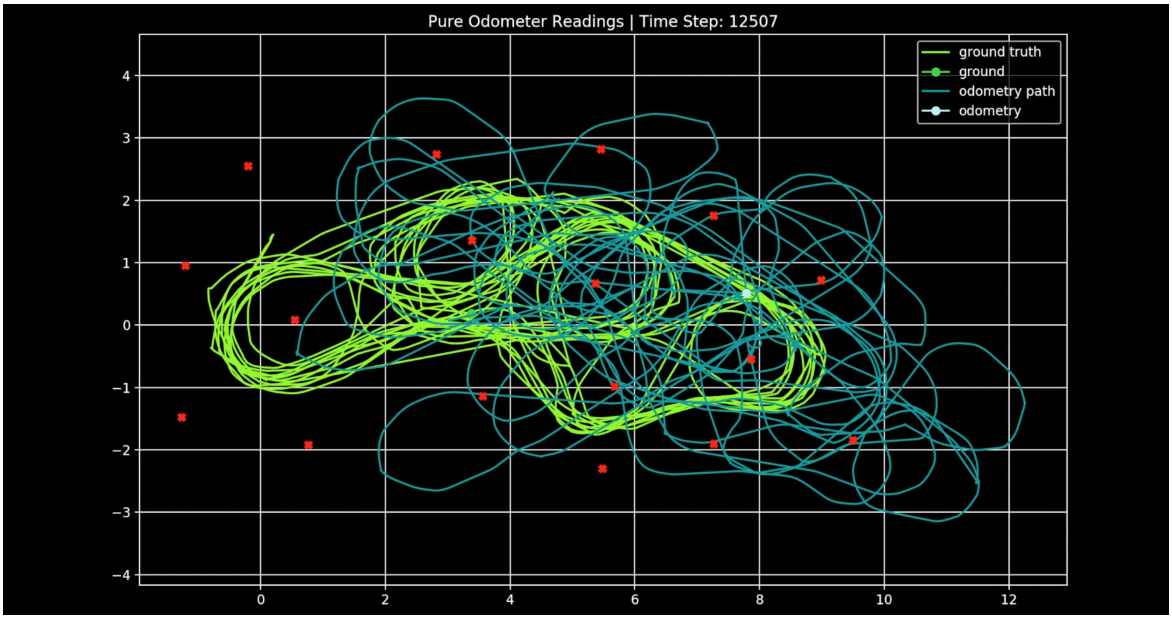

Figure 1: Odometry based trajectory

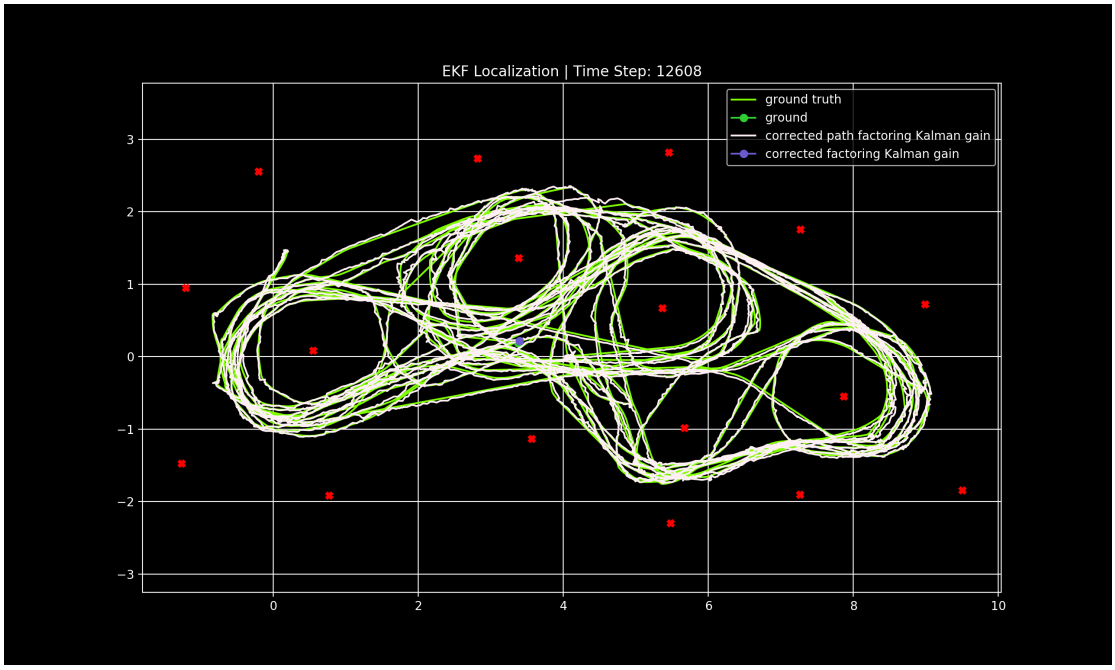

Figure 2: Corrected trajectory using Extended Kalman Filter

3 Instructions

Run the following instructions in order to execute the script.

```
python3 ekf.py
// Runs the script and live plot
```

4 Bonus question

The video for the real-time localization of the robot has been uploaded in the following link

`https://drive.google.com/drive/folders/1ElYShNr5R5cF-zP2QASaGl_r1tn9m-wY?usp=sharing`

5 Contribution

Both teammates have contributed equally in both the questions. Both the teammates have researched on the topics, required for completing the assignment. Different parts of the assignment were coded and debugged equally by each of the teammates.

$$\begin{bmatrix} x_R \\ y_R \\ \theta_R \end{bmatrix} = \begin{bmatrix} x_{R-1} \\ y_{R-1} \\ \theta_{R-1} \end{bmatrix} + dt \begin{bmatrix} v_R \cos\theta_{R-1} \\ v_R \cdot \sin\theta_{R-1} \\ \omega_R \end{bmatrix}$$

$$\frac{\delta}{\delta x_{R-1}} \begin{pmatrix} x_R \\ y_R \\ \theta_R \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + dt \begin{bmatrix} 0 & 0 & -v_R \sin\theta_{R-1} \\ 0 & 0 & v_R \cos\theta_{R-1} \\ 0 & 0 & 0 \end{bmatrix}$$

$$G = \begin{bmatrix} 1 & 0 & -dt \cdot v_R \cdot \sin\theta_{R-1} \\ 0 & 1 & dt \cdot v_R \cdot \cos\theta_{R-1} \\ 0 & 0 & 1 \end{bmatrix}$$

Observation

$$\begin{bmatrix} r_R^l \\ \phi_R^l \end{bmatrix} = \begin{bmatrix} \sqrt{(x_l - x_R - d\cos\theta_R)^2 + (y_l - y_R - d\sin\theta_R)^2} \\ \text{atan2}(y_l - y_R - d\sin\theta_R, x_l - x_R - d\cos\theta_R) - \theta_R \end{bmatrix}$$

$$\frac{\delta r_R^l}{\delta x_R} = \frac{1}{2} \cdot (r_R^l)^{-1} \cdot (-2)(x_l - x_R - d\cos\theta)$$

$$\frac{\delta r_R^l}{\delta y_R} = \frac{1}{2} \cdot (r_R^l)^{-1} \cdot (-2)(y_l - y_R - d\sin\theta_R)$$

$$\frac{\delta r_R^l}{\delta \theta_R} = \frac{1}{2} \cdot (r_R^l)^{-1} \cdot [2(x_l - x_R - d\cos\theta_R) d\sin\theta_R + (y_l - y_R - d\sin\theta_R)(-d\cos\theta_R)](2)$$

$$Q(x) = (x_l - x_R - d\cos\theta) \quad P(y) = (y_l - y_R - d\sin\theta_R)$$

$$\phi_R^l = \text{atan}\left(\frac{P(y)}{Q(x)}\right) - \theta_R$$

$$\frac{\delta \phi_R^l}{\delta x} = \frac{1}{1 + \left(\frac{P(y)}{Q(x)}\right)^2} \cdot \frac{Q(x)P'(y) - P(y)Q'(x)}{Q(x)^2}$$

$$= \frac{Q(x)^2}{Q(x)^2 + P(y)^2} \left[\frac{Q(x)P'(y) - P(y)Q'(x)}{Q(x)^2}\right]$$

$\frac{\delta}{\delta x} P(y) = 0$

$$= \frac{-P(y)Q'(x)}{Q(x)^2 + P(y)^2}$$

$$\Rightarrow Q'(x) = \frac{\delta}{\delta x_R} Q(x) = \frac{\delta}{\delta x_R}(x_l - x_R - d\cos\theta_R)$$

$$\frac{\delta}{\delta x} \phi_R^l = -\frac{P(y)(-1)}{Q(x)^2 + P(y)^2} = \frac{P(y)}{Q(x)^2 + P(y)^2}$$

$$\frac{\delta}{\delta y}\left[\text{atan}\left(\frac{P(y)}{Q(x)}\right) - \theta_R\right]$$

$\frac{\delta}{\delta y} Q(x) = 0$

$$= \frac{Q(x)^2}{Q(x)^2 + P(y)^2} \cdot \left[\frac{P'(y)Q(x) - Q'(x)P(y)}{Q(x)^2}\right]$$

$$= \frac{P'(y)Q(x)}{Q(x)^2 + P(y)^2} = \frac{-Q(x)}{Q(x)^2 + P(y)^2}$$

$$\frac{\delta}{\delta y_R}(y_l - y_R - d\sin\theta_R) = -1$$

$$\frac{\delta}{\delta \theta_R}\left[\text{atan}\left(\frac{P(y)}{Q(x)}\right) - \theta_R\right] =$$

$$= \frac{Q(x)^2}{Q(x)^2 + P(y)^2} \cdot \left[\frac{P(y)Q(x) - Q'(x)P(y)}{Q(x)^2}\right] - 1$$

$$\frac{\delta}{\delta \theta_R} Q(x) = +d\sin\theta_R$$

$$\frac{\delta}{\delta \theta_R} P(y) = -d\cos\theta_R$$

$$\Rightarrow \frac{+d\left[-\cos\theta_R Q(x) - \sin\theta_R P(y)\right]}{Q(x)^2 + P(y)^2} - 1$$

$$H = \begin{bmatrix} \frac{\delta r_1^l}{\delta x_v} & \frac{\delta r_1^l}{\delta y_v} & \frac{\delta r_1^l}{\delta \theta_v} \\ \frac{\delta \phi_1^l}{\delta x_v} & \frac{\delta \phi_1^l}{\delta y_v} & \frac{\delta \phi_1^l}{\delta \theta_v} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \end{bmatrix}$$

dimension

$2l \times 3$

l = #landmarks