Universitat de Girona
COMPUTER VISION AND ROBOTICS RESEARCH INSTITUTE (VICOROB)

Probabilistic Robotics

Lab Report

Extended Kalman FIlter Algorithm

No.	Full Name	Student ID
1	Moses Chuka Ebere	u1985468
2	Joseph Oloruntoba Adeola	u1988552

Instructor: Miguel Castillon

UDG, DECEMBER 2022

1 Objective

The aim of this lab is to implement the Extended Kalman Filter map-based algorithm to localize a Turtlebot.

2 Introduction

The Extended Kalman Filter is used when an object's motion follows a nonlinear state equation or when the measurements are nonlinear functions of the state. The linearization of the state equation and measurement equation is the foundation for the development of an extended Kalman filter. Linearization allows us to propagate the state and state covariance in a roughly linear structure, and it requires Jacobians of the state equation and measurement equation. In this lab, the Extended Kalman Filter utilizes a series of detections or measurements to estimate the state of the robot based on the robot's motion model. In the motion model, the state is a collection of quantities that represent the status of the turtlebot, in our case, the turtlebot's position (x, y) and heading θ.

3 State Definition

The state of turtlebot at time step t in is given by

$$x_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}$$

4 Process model

As the robot moves, there's a need to always compute its new state. To achieve this, the displacement measured by the odometry (including the noise) in the robot frame is compounded with the previous state of the robot.

$Next\_robot\_state = Previous\_robot\_position + Odometry\_displacement$

But the robot's state and the odometry measurement are in two different frames, thus, the odometry measurement has to be transformed to the world frame W, before adding it to the robot's previous state. In this case, the transformation matrix is a rotation matrix about the z-axis given by:

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Suppose x_{t-1}^W is the robot's previous state in, u_t^r the odometry displacement at time t, and w_t^r, the uncertainty (noise) in the odometry measurement.

where

$$x_{t-1}^W = \begin{bmatrix} x_{t-1}^W \\ y_{t-1}^W \\ \theta_{t-1}^W \end{bmatrix}, \quad u_t^r = \begin{bmatrix} x_t^r \\ y_t^r \\ \theta_t^r \end{bmatrix}, \text{ and } \; w_t^r = \begin{bmatrix} w_{x_t}^r \\ w_{y_t}^r \\ w_{\theta t}^r \end{bmatrix}$$

,

The robot's new state x_t^W is

$$x_t^W = x_{t-1}^W \oplus (u_t^r + w_t^r) = \begin{bmatrix} x_{t-1}^W \\ y_{t-1}^W \\ \theta_{t-1}^W \end{bmatrix} + \left\{ \begin{matrix} \begin{bmatrix} \cos\theta_{t-1} & -\sin\theta_{t-1} \\ \sin\theta_{t-1} & \cos\theta_{t-1} \end{bmatrix} \begin{bmatrix} x_t^r + w_{x_t}^r \\ y_t^r + w_{y_t}^r \end{bmatrix} \\ \theta_{t-1}^r + w_{\theta t}^r \end{matrix} \right\}$$

Since this equation contains trigonometric identities, it is a non-linear equation. Thus, the non-linear process model can be written as

$$f(x_{t-1}^W, u_t^r, w_t^r) = x_{t-1}^W \oplus (u_t^r + w_t^r)$$

5 Implementation

The Extended Kalman Filter algorithm is divided into three steps. The prediction, data association, and update. These three steps have been implemented in the python file submitted alongside this report and the algorithm procedure discussed in subsequent sections.

5.1 EKF Prediction

The prediction step can be divided into two steps. First is the computation of the predicted new state while the second is the computation of the new covariance.

$$\begin{aligned} x_{k|k-1} &= f(x_{k-1}, u_k, w_k) \\ P_{k|k-1} &= A_k P_{k-1} A_k^T + W_k Q_k W_k^T \end{aligned}$$

The first step, i.e, prediction of the robot's new state is computed using the last equation (f) earlier derived in the process model section. While the new $P_{k|k-1}$ covariance is derived as follows. The Matrix A_k and W_k are Jacobians derived from the function $f(x_{k-1}, u_k, w_k)$. A_k is the Jacobian of the predicted f state with respect to the previous state and W_k is the Jacobian of the predicted state f with respect to the noise.

$$A_k = \frac{\partial f(x_{k-1}, u_k, w_k)}{\partial x_{k-1}} = \begin{bmatrix} \frac{\partial f_x}{\partial x_{k-1}} & \frac{\partial f_x}{\partial y_{k-1}} & \frac{\partial f_x}{\partial \theta_{k-1}} \\ \frac{\partial f_y}{\partial x_{k-1}} & \frac{\partial f_y}{\partial y_{k-1}} & \frac{\partial f_y}{\partial \theta_{k-1}} \\ \frac{\partial f_\theta}{\partial x_{k-1}} & \frac{\partial f_\theta}{\partial y_{k-1}} & \frac{\partial f_\theta}{\partial \theta_{k-1}} \end{bmatrix}$$

where A_k is evaluated at $w_k = 0$.
But

$$f(x_{k-1}, u_k, w_k) = \begin{bmatrix} x_{k-1}^W \\ y_{k-1}^W \\ \theta_{k-1}^W \end{bmatrix} + z \left\{ \begin{matrix} \begin{bmatrix} \cos\theta_{k-1} & -\sin\theta_{k-1} \\ \sin\theta_{k-1} & \cos\theta_{k-1} \end{bmatrix} \begin{bmatrix} x_k^r + w_{x_k}^r \\ y_k^r + w_{x_k}^r \end{bmatrix} \\ \theta_{k-1}^r + w_{\theta k}^r \end{matrix} \right\}$$

$$f(x_{k-1}, u_k, w_k) = \begin{bmatrix} x_{k-1}^W + (x_k^r + w_{x_k}^r)\cos\theta_{k-1} - (y_k^r + w_{y_k}^r)\sin\theta_{k-1} \\ y_{k-1}^W + (x_k^r + w_{x_k}^r)\sin\theta_{k-1} + (y_k^r + w_{y_k}^r)\cos\theta_{k-1} \\ \theta_{k-1}^W + \theta_{k-1}^r + w_{\theta k}^r \end{bmatrix}$$

Thus,

$$A_k = \frac{\partial f(x_{k-1}, u_k, w_k)}{\partial x_{k-1}} = \begin{bmatrix} 1 & 0 & -(x_k^r + w_{x_k}^r)\sin\theta_{k-1} - (y_k^r + w_{y_k}^r)\cos\theta_{k-1} \\ 0 & 1 & (x_k^r + w_{x_k}^r)\cos\theta_{k-1} - (y_k^r + w_{y_k}^r)\sin\theta_{k-1} \\ 0 & 0 & 1 \end{bmatrix}$$

at $w_k = 0$

$$A_k = \frac{\partial f(x_{k-1}, u_k, w_k)}{\partial x_{k-1}} = \begin{bmatrix} 1 & 0 & -(x_k^r)\sin\theta_{k-1} - (y_k^r)\cos\theta_{k-1} \\ 0 & 1 & (x_k^r)\cos\theta_{k-1} - (y_k^r)\sin\theta_{k-1} \\ 0 & 0 & 1 \end{bmatrix}$$

where superscripts W and r are the world and robot's reference frame respectively. In like manner,

$$W_k = \frac{\partial f(x_{k-1}, u_k, w_k)}{\partial w_k} = \begin{bmatrix} \frac{\partial f_x}{\partial w_x} & \frac{\partial f_x}{\partial w_y} & \frac{\partial f_x}{\partial w_\theta} \\ \frac{\partial f_y}{\partial w_x} & \frac{\partial f_y}{\partial w_y} & \frac{\partial f_y}{\partial w_\theta} \\ \frac{\partial f_\theta}{\partial w_x} & \frac{\partial f_\theta}{\partial w_y} & \frac{\partial f_\theta}{\partial w_\theta} \end{bmatrix}$$

On differentiating,

$$W_k = \begin{bmatrix} \cos\theta_{k-1} & -\sin\theta_{k-1} & 0 \\ \sin\theta_{k-1} & \cos\theta_{k-1} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

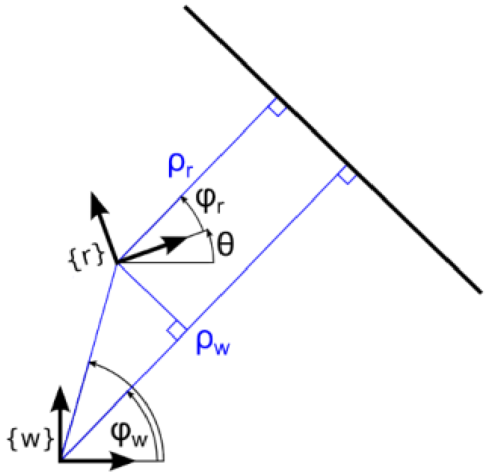
And Q_k which is the noise in the odometry is defined in the lab guide as:

$$Q_k = \begin{bmatrix} 0.025 & 0 & 0 \\ 0 & 0.025 & 0 \\ 0 & 0 & deg2rad(2) \end{bmatrix}$$

The off-diagonal zero elements in Qk show that the noise are uncorrelated. Thus, using the derived Ak, Wk, Qk and the initial covariance P_{k-1}, $P_{k|k-1}$ can be calculated

5.2 Data Association

In the data association step, the algorithm tries to match the line measurements taken by the sensors of the robot and compares these lines with the map of the environment. To be able to carry out this comparison, a measurement model was derived as described below.



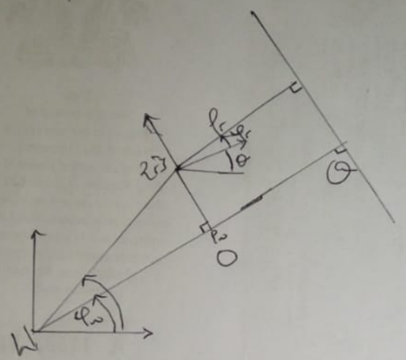
The robot has range and bearing sensors so it measures the range and bearing, (ρ_r, φ_r) of each line it senses in the environment. The map lines are all in cartesian coordinates, thus they need to be converted to the range and bearing. After conversion, these lines will then be transformed into the robot's frame before comparing the measured line with all the map lines. The expected measurement function $h(x_k|x_{k-1})$ that transforms a feature from the real map in polar coordinates in the world frame ${}^W x_f$ to a feature in polar coordinates in the robot frame ${}^R x_f$ as shown in the figure above is given by

$$h(x_k|x_{k-1}) = \begin{bmatrix} \rho_r \\ \varphi_r \end{bmatrix} \begin{bmatrix} \rho_r - x\cos\varphi_w - y\sin\varphi_w \\ \varphi_w - \theta \end{bmatrix}$$

The proof is presented below

From the figure in

Our goal is to transform a polar co-ordinate in the world frame to polar co-ordinate in the robot's reference frame



We are looking for ρ_w first.

~~from~~ Let $\rho_w = WOQ$

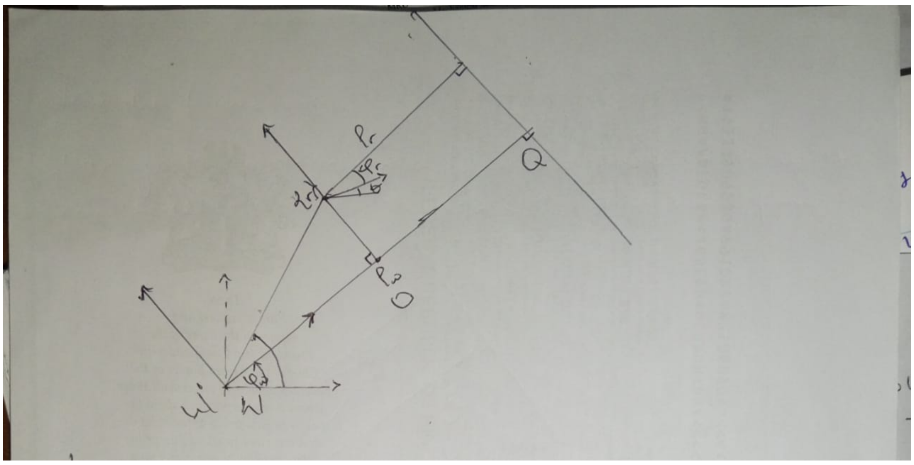
where $OQ = \rho_r$

The line $\rho_w = WO + \rho_r$

To get the length of WO,

we transform the world frame by $-\varphi_w$ about the z-axis $\{W\}$ and project the robot's frame $\{r\}$ in W.

Doing this, we'll get



A point $({}^{w}x, {}^{w}y)$ in the world frame can be found in the w' frame through

$$\begin{bmatrix} {}^{w}x \\ {}^{w}y \end{bmatrix} = \begin{pmatrix} \cos(-\varphi_w) & -\sin(-\varphi_w) \\ \sin(-\varphi_w) & \cos(-\varphi_w) \end{pmatrix} \begin{pmatrix} {}^{w'}x \\ {}^{w'}y \end{pmatrix}$$

Assuming $({}^{w'}x, {}^{w'}y)$ is the co-ordinate of frame $\{r\}$ in the new frame w', then

$$\begin{bmatrix} {}^{w}x \\ {}^{w}y \end{bmatrix} = \begin{pmatrix} \cos\varphi_w & \sin\varphi_w \\ -\sin\varphi_w & \cos\varphi_w \end{pmatrix} \begin{pmatrix} {}^{w'}x \\ {}^{w'}y \end{pmatrix} = \begin{pmatrix} {}^{w'}x\cos\varphi_w + {}^{w'}y\sin\varphi_w \\ -{}^{w'}x\sin\varphi_w + {}^{w'}y\cos\varphi_w \end{pmatrix}$$

$$\begin{bmatrix} {}^{w}x \\ {}^{w}y \end{bmatrix} = \begin{bmatrix} {}^{w'}x\cos\varphi_w + {}^{w'}y\sin\varphi_w \\ -{}^{w'}x\sin\varphi_w + {}^{w'}y\cos\varphi_w \end{bmatrix}$$

But the line WD that we are looking for is the x co-ordinate of frame $\{r\}$ in W', therefore

$$WD = {}^{w'}x\cos\varphi_w + {}^{w'}y\sin\varphi_w \quad \text{— (2)}$$

substituting eqn (2) into eqn (1) we obtain

$$\rho_w = {}^{w'}x\cos\varphi_w + {}^{w'}y\sin\varphi_w + \rho_r \quad \text{— (3)}$$

making ρ_r subject of the formula.

$$\boxed{\rho_r = \rho_w - {}^{w'}x\cos\varphi_w - {}^{w'}y\sin\varphi_w}$$

which is the range transformed from the world frame to the robot frame.

* The heading transformed from the world frame to the robot frame is simply

$$\boxed{\varphi_r = \varphi_w - \theta}$$

Deduced geometrically from the figure.

While comparing the measured line with all the lines on the map, we estimate the Mahalanobis distance, and the smallest distance is kept. We then check then compare this distance with a set threshold. If the smallest Mahalanobis distance is satisfies this threshold, we consider the line that gave this distance. The data association step is implemented through the algorithm stated below:

$$
\begin{aligned}
&v_{ij} = z_i - h(x_{k|k-1}) \\
&S_{ij} = H_j P_{k|k-1} H_j^T + R_i \\
&D_{ij}^2 = v_{ij}^T S_{ij}^{-1} v_{ij} \\
&\text{Take the smallest } D_{ij}^2 \text{ if } D_{ij}^2 < \chi^2_{\rho,\varphi}
\end{aligned}
$$

The matrix H, which is the Jacobian of the measurement with respect to the state is given by:

$$
H = \frac{\partial h}{\partial x} = \begin{bmatrix} \frac{\partial h_\rho}{\partial x} & \frac{\partial h_\rho}{\partial y} & \frac{\partial h_\rho}{\partial \theta} \\ \frac{\partial h_\varphi}{\partial x} & \frac{\partial h_\varphi}{\partial y} & \frac{\partial h_\varphi}{\partial \theta} \end{bmatrix} = \begin{bmatrix} -cos\varphi_w & -sin\varphi_w & 0 \\ 0 & 0 & -1 \end{bmatrix}
$$

Similarly, R_k, the covariance matrix for the observations and the angle measurement noise has been estimated to 10 degrees and the range measurement to 0.2 m. R_k matrix is estimated as constant and is given by:

$$
R_k = \begin{bmatrix} 0.2 & 0 \\ 0 & deg2rad(10) \end{bmatrix}
$$

5.3 Update

The update step is the final step of the Extended Kalman Filter Algorithm. Here, the Kalman gain K_k, the new mean x_k, and the new covariance P_k are calculated. This step was implemented by simply substituting the Jacobians already calculated in the data association step into the algorithm.

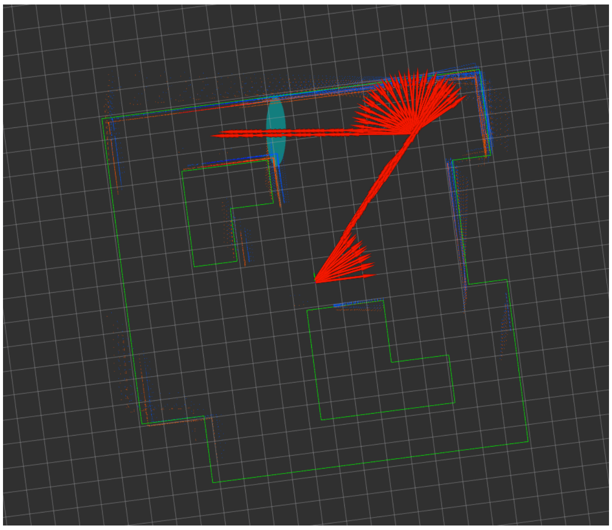
$$
\begin{aligned}
&v_k = z_k - h(x_{k|k-1}) \\
&S_k = H_k P_{k|k-1} H_k^T + R_k \\
&K_k = P_{k|k-1} H_k^T S_k^{-1} \\
&x_k = x_{k|k-1} + K_k v_k \\
&P_k = (I - K_k H_k)\, P_{k|k-1}\, (I - K_k H_k)^T + K_k R_k K_k^T
\end{aligned}
$$

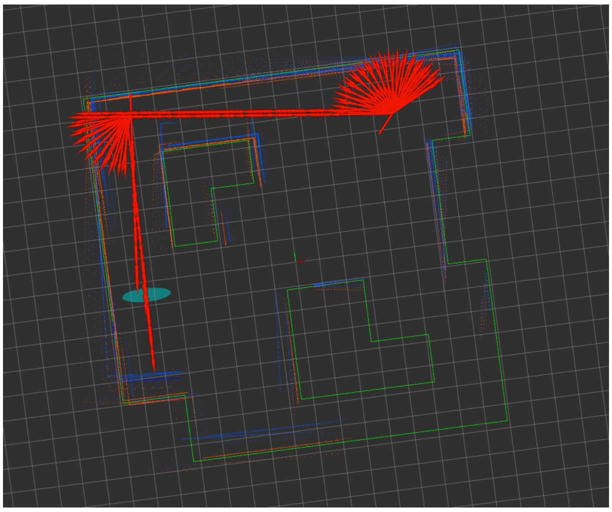
5.4 Optional Part

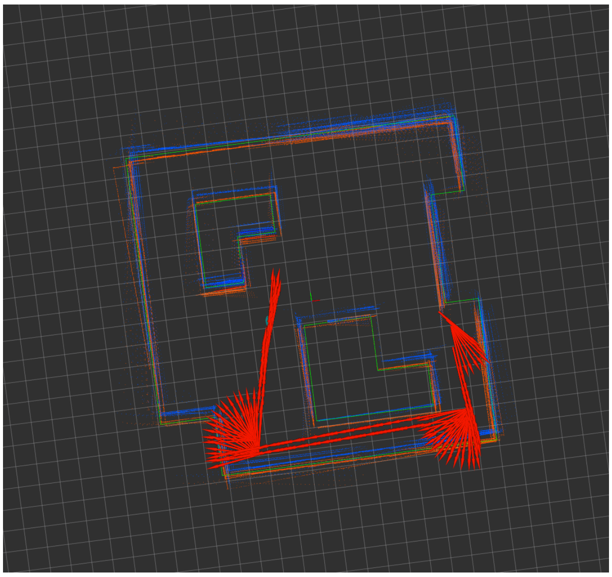
The optional part of the lab was implemented in the data association step by comparing the length of each measured line with the length of all the lines in the environment map. This length was calculated using the euclidean distance formula. If the length of the measured line is greater than the length of all the lines in the map, the variable *islonger* will be set to true else it would be set to false.

6 Results

On successfully implementing the algorithm, the figures below show the results obtained as the robot moves in the environment. The uncertainty increases in width when the robot moves in the x-direction and increases in height when the robot moves in the y-direction. When the update step runs, the uncertainty reduces in size as the robot becomes more confident in its belief.







7 Discussion

The major challenge encountered while implementing the EKF algorithm was in deriving the measurement model because it requires a thorough understanding of the problem before one can correctly derive the measurement equations.