1 Introduction

In this example we introduce the famous and widely used Extended Kalman Filter (EKF) which is used in estimation theory for nonlinear models and non-Gaussian random variables. We begin by introducing the problem setup for recursive discrete time estimation and follow by definitions and implementations of the of the Extended Kalman Filter in a 2D localization problem.

1.1 Nonlinear, Non Gaussian Recursive Discrete-Time Estimation

In the Linear Gaussian estimation case the mean and mode of the posteriors are the same which led to equivalent results for maximum a posterior and full Bayesian solutions. This statement does not hold true for nonlinear non Gaussian (NLNG) case. Just as in the KF example we define motion and observation models:

$$\text{motion model:} \quad \mathbf{x}_k = \mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{v}_k, \mathbf{w}_k\right), \quad k = 1 \ldots K \tag{1}$$

$$\text{observation model:} \quad y_k = g\left(x_k, n_k\right), \quad k = 0 \ldots K \tag{2}$$

Where we have f(·) as the nonlinear motion model and the function g(·) as the nonlinear observation model. The Bayes filter was then introduced as a base template for the derivations of the EKF and many more such as the Particle Filter, and (iterated) sigmapoint Kalman filter. The difference is determined by which approximations are made from the original Bayes Filter i.e.(approximating the probability density functions(PDF) as Gaussian, or approximating the PDFs using a large number of samples).

1.2 Extended Kalman Filter

We now show that if the belief is constrained to be Gaussian, the noise is Gaussian, and we linearize the motion and observation models in order to carry out the integral in the Bayes filter, we will arrive at the famous (EKF). We begine by linearzing the motion and observation models about the current estimate.

$$\mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{v}_k, \mathbf{w}_k\right) \approx \check{\mathbf{x}}_k + \mathbf{F}_{k-1}\left(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}\right) + \mathbf{w}'_k \tag{3}$$

$$\mathbf{g}\left(\mathbf{x}_k, \mathbf{n}_k\right) \approx \check{\mathbf{y}}_k + \mathbf{G}_k\left(\mathbf{x}_k - \check{\mathbf{x}}_k\right) + \mathbf{n}'_k \tag{4}$$

Where:

$$\check{\mathbf{x}}_k = \mathbf{f}\left(\hat{\mathbf{x}}_{k-1}, \mathbf{v}_k, \mathbf{0}\right) \tag{5}$$

$$\mathbf{F}_{k-1} = \left.\frac{\partial \mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{v}_k, \mathbf{w}_k\right)}{\partial \mathbf{x}_{k-1}}\right|_{\hat{\mathbf{x}}_{k-1}, \mathbf{v}_k, \mathbf{o}} \tag{6}$$

$$\mathbf{w}_k' = \left.\frac{\partial \mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{v}_k, \mathbf{w}_k\right)}{\partial \mathbf{w}_k}\right|_{\hat{\mathbf{x}}_{k-1}, \mathbf{v}_k, \mathbf{o}} \mathbf{w}_k \tag{7}$$

$$\check{\mathbf{y}}_k = \mathbf{g}\left(\check{\mathbf{x}}_k, \mathbf{0}\right) \tag{8}$$

$$\mathbf{G}_k = \left.\frac{\partial \mathbf{g}\left(\mathbf{x}_k, \mathbf{n}_k\right)}{\partial \mathbf{x}_k}\right|_{\overline{\mathbf{x}}_k, \mathbf{o}} \tag{9}$$

$$\mathbf{n}_k' = \left.\frac{\partial \mathbf{g}\left(\mathbf{x}_k, \mathbf{n}_k\right)}{\partial \mathbf{n}_k}\right|_{\hat{\mathbf{x}}_k, \mathbf{0}} \mathbf{n}_k \tag{10}$$

By using the statistical properties of the current state and measurements, and substituting these results into the Bayes filter we can result to the final representation of the EKF.

Prediction Steps:

$$\dot{\mathbf{P}}_k = \mathbf{F}_{k-1} \hat{\mathbf{P}}_{k-1} \mathbf{F}_{k-1}^T + \mathbf{Q}_k' \tag{11}$$

$$\check{\mathbf{x}}_k = \mathbf{f}\left(\hat{\mathbf{x}}_{k-1}, \mathbf{v}_k, \mathbf{0}\right) \tag{12}$$

Kalman Gain:

$$\mathbf{K}_k = \check{\mathbf{P}}_k \mathbf{G}_k^T \left(\mathbf{G}_k \dot{\mathbf{P}}_k \mathbf{G}_k^T + \mathbf{R}_k'\right)^{-1} \tag{13}$$

Correction Steps:

$$\hat{\mathbf{P}}_k = \left(\mathbf{1} - \mathbf{K}_k \mathbf{G}_k\right) \check{\mathbf{P}}_k \tag{14}$$

$$\hat{\mathbf{x}}_k = \check{\mathbf{x}}_k + \mathbf{K}_k \left(\mathbf{y}_k - \mathbf{g}\left(\check{\mathbf{x}}_k, \mathbf{0}\right)\right) \tag{15}$$

There is no guarantee that the EFK will perform well for a general nonlinear system. The EKF does not alter the Kalman filter's linear equations. Instead, it linearizes the nonlinear equations at the mean of the estimate of the state, and uses this linearization in the linear Kalman filter. Lastly, we also note that there are Jacobians embedded in the covariances for the noise [1].

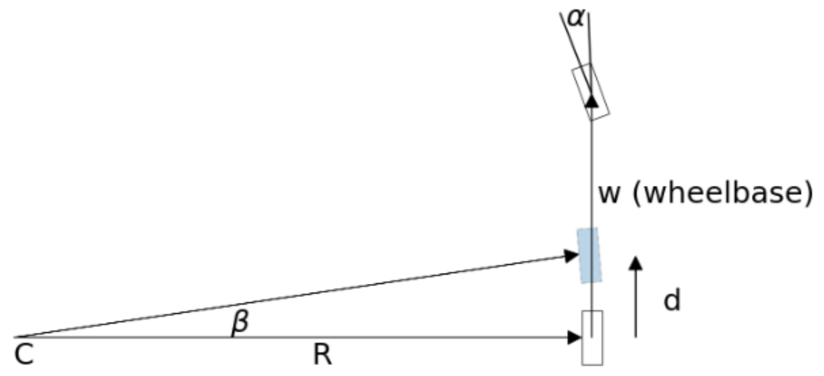

Figure 1: Simple Bicycle Steering Model [2].

2 Implementation and Results

We first introduce the bicycle steering model whose equations we can use as an approximation of an automobile steering system, which in reality faced complicated issues such as slippage, and is accurately modelled by a complicated set of differential equations.

Where we have:

$$\begin{aligned}
&\beta = \frac{d}{w}\tan(\alpha)\\
&x = x - R\sin(\theta) + R\sin(\theta + \beta)\\
&y = y + R\cos(\theta) - R\cos(\theta + \beta)\\
&\theta = \theta + \beta
\end{aligned}$$

We begin by introducing the state and control input as :

$$\mathbf{x}_k = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{16}$$

The control input as:

$$\mathbf{v}_k = \begin{bmatrix} \nu \\ \alpha \end{bmatrix} \tag{17}$$

Where ν and α are the velocity and steering angle in m/s and radians.

From equation 5, we can expand the robot motion model as:

$$\mathrm{x}_k = \mathbf{x}_{k+1} + \left[\begin{array}{c} -R\sin(\theta) + R\sin(\theta+\beta) \\ R\cos(\theta) - R\cos(\theta+\beta) \\ \beta \end{array} \right] (18)$$

We then find F by taking the jacobian of $\mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{v}_k, \mathbf{w}_k\right)$ with respect to the state parameters of the previous timestep. We obtain:

$$\mathbf{F}_{k-1} = \left[\begin{array}{ccc} 1 & 0 & -R\cos(\theta) + R\cos(\theta+\beta) \\ 0 & 1 & -R\sin(\theta) + R\sin(\theta+\beta) \\ 0 & 0 & 1 \end{array} \right] \tag{19}$$

We let $\mathbf{Q}_k' = \mathbf{V}\mathbf{M}\mathbf{V}^{\mathrm{T}}$, where:

$$\mathbf{M} = \left[\begin{array}{cc} \sigma_{\text{vel}}^2 & 0 \\ 0 & \sigma_\alpha^2 \end{array} \right] \tag{20}$$

$$\mathbf{V} = \frac{\partial f(x,u)}{\partial u} \left[\begin{array}{cc} \frac{\partial f_1}{\partial y_2} & \frac{\partial f_1}{\partial q_2} \\ \frac{\partial f_3}{\partial y_3} & \frac{\partial f_3}{\partial \alpha} \end{array} \right] \tag{21}$$

We now design the measurement model of the system. If P is the position of the landmark we can define the range as:

$$r = \sqrt{\left(p_x - x\right)^2 + \left(p_y - y\right)^2} \tag{22}$$

and the bearing of the sensor reading as:

$$\phi = \arctan\left(\frac{p_y - y}{p_x - x}\right) - \theta \tag{23}$$

Thus our measurement model in equation 8 can then we written as:

$$\left[\begin{array}{c} \sqrt{\left(p_x - x\right)^2 + \left(p_y - y\right)^2} \\ \arctan\left(\frac{p_y - y}{p_x - x}\right) - \theta \end{array} \right] \tag{24}$$

This expression is nonlinear. So we need to linearize this nonlinear measurement model by taking it's jacobian at the current state. Thus equation 9 takes the form:

$$\left[\begin{array}{ccc} \frac{-p_y + x}{\sqrt{(p_x - x)^2 + (p_y - y)^2}} & \frac{-p_y + y}{\sqrt{(p_x - x)^2 + (p_y - y)^2}} & 0 \\ -\frac{-p_y + y}{(p_x - x)^2 + (p_y - y)^2} & -\frac{p_x - x}{(p_x - x)^2 + (p_y - y)^2} & -1 \end{array} \right] \tag{25}$$

Finally, we can assume that the noise of the range and bearing measurements are independent and hence:

$$\mathbf{R}_k' = \left[\begin{array}{cc} \sigma_{\text{range}}^2 & 0 \\ 0 & \sigma_{\text{bearing}}^2 \end{array} \right] \tag{26}$$

SymPy, which is a library for symbolic mathematics was used to compute the jacobians. In this problem we want our robot to move in a circular trajectory around a landmark located at [10,10]. The radius of the trajectory is given by the formula $R = \frac{w}{\tan(\alpha)}$, where w is the wheelbase and α is the steering angle as mentioned earlier. In this problem our robot has a wheelbase of 0.5 m, steering angle of 0.0499 radians, velocity of 5 m/s, standard deviation of velocity as 1.e-10, standard deviation of the steering angle as 1.e-10, standard deviation of the range measurement as 1.4 and lastly a standard deviation for the bearing measurement as 0.05. The robots initial pose is $[10, 0, 0]$ and the initial state covarience is $[0.1, 0.1, 0.1]$. These values were scaled accordingly for much more recognizable animation to be visualized. The prediction step occurs 8 times a second following a measurement step. As in the case with the KF, after the measurement takes place the co variance of the state belief decreases, as the measurements provide us with more certainty. Given the radius of trajectory, and the robots velocity, we conclude that after approximately 12.57 seconds our robot will move thorough one full circle of radius 10.01 m around the landmark [10,10]. In the graphical trajectory model of Figure 2, the grey eclipses represent the covariance for the state after each prediction step, and the center of the eclipse is located at the x and y coordinates of where the robot is estimated to be located at that given time based on the computations from the predication steps only. The same goes for the green eclipses which correspond to the correctional steps and thus incorporate the measurements from the landmark. A snippet of the animation is shown in Figure 3. The solid red line trajectory represents the predicted trajectory without the measurements, and dotted green line represents the corrected trajectory of the robot.

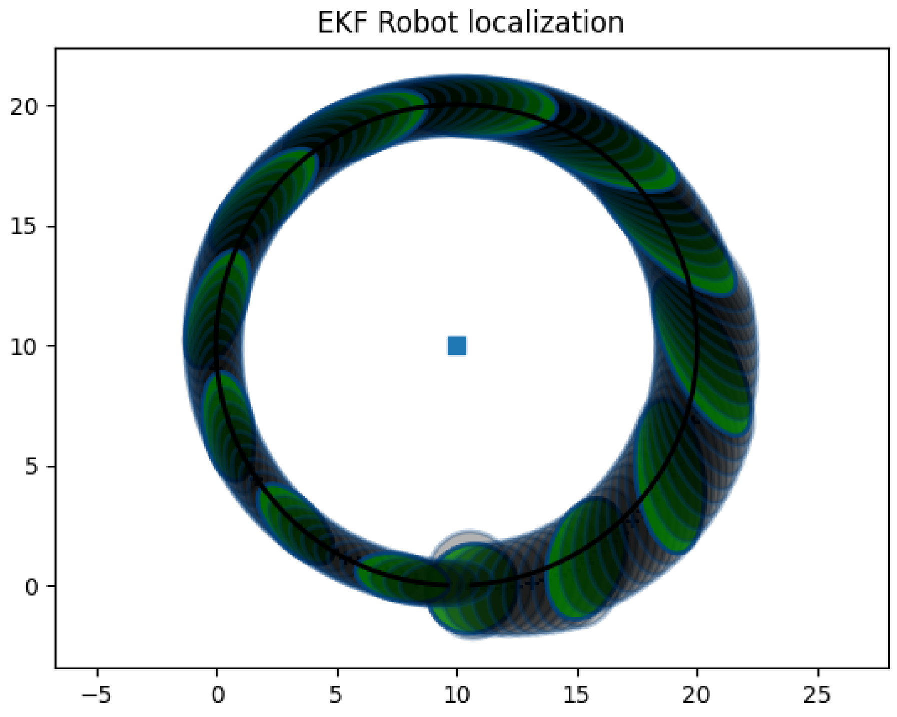

Figure 2: Graphical Trajectory Model

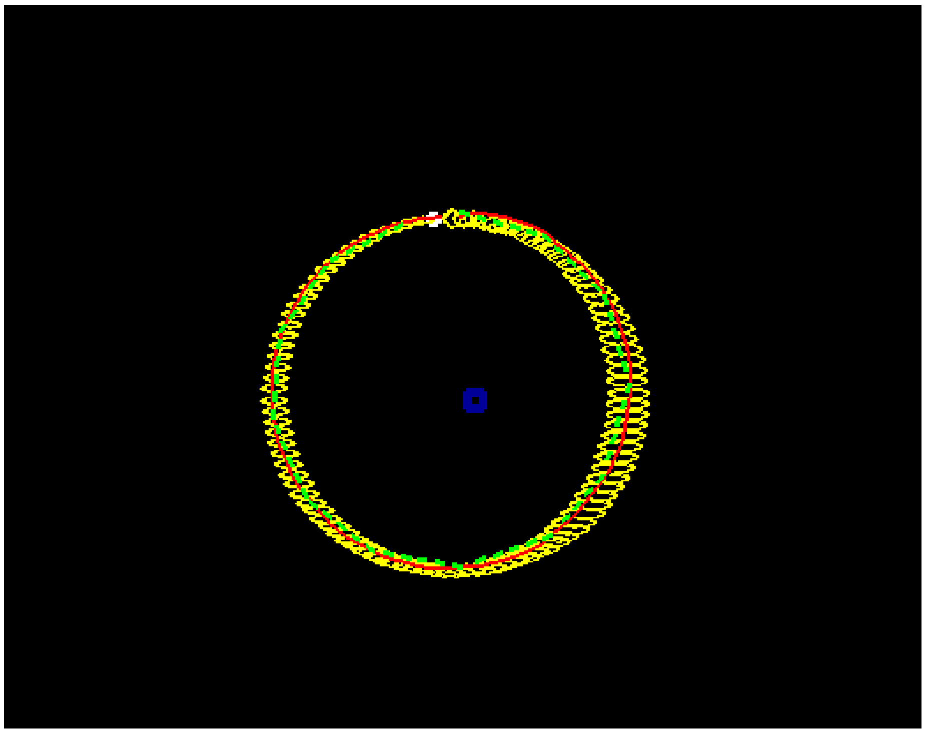

Figure 3: Snippet of Animated Trajectory

References

[1] T. D. Barfoot, *State Estimation for Robotics.* Cambridge University Press, 2020.

[2] R. R. L. Jr, *Kalman and Bayesian Filters in Python*, 2020.