The Extended Kalman Filter

Mehdi Raza Khorasani, Habib University 26/6/2021

Introduction

The extended Kalman Filter is an extension to the Kalman Filter Algorithm. The KF algorithm is defined for Discrete time Linear time invariant systems (DT LTI), which are of the form:

$$x_k = Fx_{k-1} + V_k \tag{1}$$

$$y_k = Hx_k + W_k \tag{2}$$

The algorithm fails when the system cannot be represented as (1) and (2). The extended KF approach suggests that we linearize the system and obtain a linear approximation, and then develop a KF algorithm for it. The following lines describe how a non linear system can be linearized and a KF algorithm can be applied to it.

The discrete time non-linear system

Consider a non-linear system, defined by the following equations:

$$x_k = f(x_k) + V_k \tag{3}$$

$$y_k = h(x_k) + W_k \tag{4}$$

where, V_k and W_k is white uncorrelated Gaussian noise defined as follows:

$$V_k \sim (0, Q_k)$$

$$W_k \sim (0, R_k)$$

The EKF Algorithm

The algorithm of the EKF consists of linearlizing the plant about the optimal value and consecutively applying the baysian estimation steps. Similarly, the observations are also linearized about the optimal value. The following lines summarize the filtering algorithm:

1. Linearization: Compute the Jacobian of f at $\hat{x}_{k-1|k-1}$ i.e. about the last optimal estimate of x_k:

$$F_k = \nabla_{X^T} f(x) \mid_{x=\hat{x}_{k-1|k-1}}$$

2. State Prediction: Compute Predicted mean and co-variance matrix:

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1})$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$$

3. Linearization: Compute the jacobian of h at $\hat{x}_{k|k-1}$ i.e. about the estimate of x given the last estimate:

$$H_k = \nabla_{X^T} h(x) \mid_{x=\hat{x}_{k|k-1}}$$

4. Measurement Prediction: Compute predicted mean, covariance and Kalman gain:

$$\hat{y}_{k|k-1} = h(\hat{x}_{k|k-1})$$

$$S_k = H_k P_{k|k-1} H_k^T + R_k$$

$$K_k = P_{k|k-1} H_k^T S_k^{-1}$$

5. Estimation: Compute the posterior mean and co-variance as follows:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_{k|k-1})$$

$$P_{k|k} = P_{k|k-1} - K_k H_k P_{k|k-1}$$

Simulation

We simulate the non-linear system observed by a Radar. The system and radar are governed by the following equations:

$$\boldsymbol{x_k} = \begin{bmatrix} x_k \\ y_k \\ \phi_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + Tv_k cos(\phi_{k-1}) \\ y_{k-1} + Tv_k sin(\phi_{k-1}) \\ \phi_{k-1} + T\omega_k \end{bmatrix} + \begin{bmatrix} V_{1,k} \\ V_{2,k} \\ V_{3,k} \end{bmatrix} \tag{5}$$

$$\boldsymbol{y_k} = \begin{bmatrix} \sqrt{x_k^2 + y_k^2} \\ tan^{-1} \frac{y_k}{x_k} \end{bmatrix} \tag{6}$$

where v_k, ω_k are the linear and angular velocities of the object respectively and T is sampling time. They are taken as constants:

$$v_k = 0.1, \omega_k = 0.01, T = 0.05$$

Algorithm

0.0.1 Initialization

1. The co-variance Matrix of process Noise was taken as follows:

$$Q_k = \begin{bmatrix} 1e^{-6} & 0 & 0 \\ 0 & 1e^{-6} & 0 \\ 0 & 0 & 1e^{-6} \end{bmatrix}$$

2. The co-variance Matrix of Radar Noise was taken as follows:

$$R_k = \begin{bmatrix} 1e^{-4} & 0 & 0 \\ 0 & 1e^{-4} & 0 \\ 0 & 0 & 1e^{-4} \end{bmatrix}$$

3. $\mathbf{MATLAB}$'s $randn()$ function was used to generate the the noise vectors V_k and W_k of dimensions 3×1 and 2×1 respectively.

4. Compute the true value of x_k and y_k from the model's equation (5) and (6) to simulate the non-linear object and Radar.

0.0.2 Linearization

1. The pseudo function for this step is as follows:

```
F_k = F_jacobian(T, vk, xhat_km1)
```

2. The jacobian of f is computed as follows:

$$F_k = \begin{bmatrix} \frac{\partial f_1}{\partial x_k} & \frac{\partial f_1}{\partial y_k} & \frac{\partial f_1}{\partial \phi_k} \\ \frac{\partial f_2}{\partial x_k} & \frac{\partial f_2}{\partial y_k} & \frac{\partial f_2}{\partial \phi_k} \\ \frac{\partial f_3}{\partial x_k} & \frac{\partial f_3}{\partial y_k} & \frac{\partial f_3}{\partial \phi_k} \end{bmatrix}_{\mathbf{x}=\hat{x}_{k-1|k-1}}$$

where:

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} x_{k-1} + T v_k \cos(\phi_{k-1}) \\ y_{k-1} + T v_k \sin(\phi_{k-1}) \\ \phi_{k-1} + T w_k \end{bmatrix}_{\mathbf{x}=\hat{x}_{k-1|k-1}}$$

substituting values and taking respective partial derivatives, we get:

$$F_k = \begin{bmatrix} 1 & 0 & -T v_k \sin \phi_{k-1} \\ 0 & 1 & T v_k \cos \phi_{k-1} \\ 0 & 0 & 1 \end{bmatrix}_{\mathbf{x}=\hat{x}_{k-1|k-1}}$$

0.0.3 State Prediction

1. Pseudo function for this step is:

```
[xhat_predict, P_predict] = state_predict(xhat_km1, P_km1, ↩
    F_k,  Q_k, vk, wk, T)
```

2. The mean is predicted by substituting $\hat{x}_{k-1|k-1}$ directly in f of equation (5)

3. The co-variance is predicted as described earlier.

0.0.4 Linearization

1. The pseudo function for this step is as follows:

```
H_k = H_jacobian(xhat_predict)
```

2. The jacobian of h is computed similar to what was described earlier. It comes out to be:

$$H_k = \begin{bmatrix} -\frac{x_k}{\sqrt{x_k^2+y_k^2}} & \frac{x_k}{\sqrt{x_k^2+y_k^2}} & 0 \\ \frac{-y_k}{x_k^2+y_k^2} & \frac{x_k}{x_k^2+y_k^2} & 0 \end{bmatrix}_{\mathbf{x}=\hat{x}_{k|k-1}}$$

0.0.5 Measurement Prediction

The pseudo function for this step is as follows:

```
[yhat_last,K_k] = measurement_predict(xhat_predict, H_k, P_predict↩
    , R_k)
```

0.0.6 Estimation

The pseudo function for this step is as follows:

```
[xhat_optimal,P_optimal] = estimate(y_k, yhat_predict, P_predict, ↩
    H_k, K_k)
```

Results

The true trajectory, sensor observations and predicted trajectory are plotted in figure 1. As can be observed, the EKF algorithm is able to rightly and accurately able to localize the non-linear object.

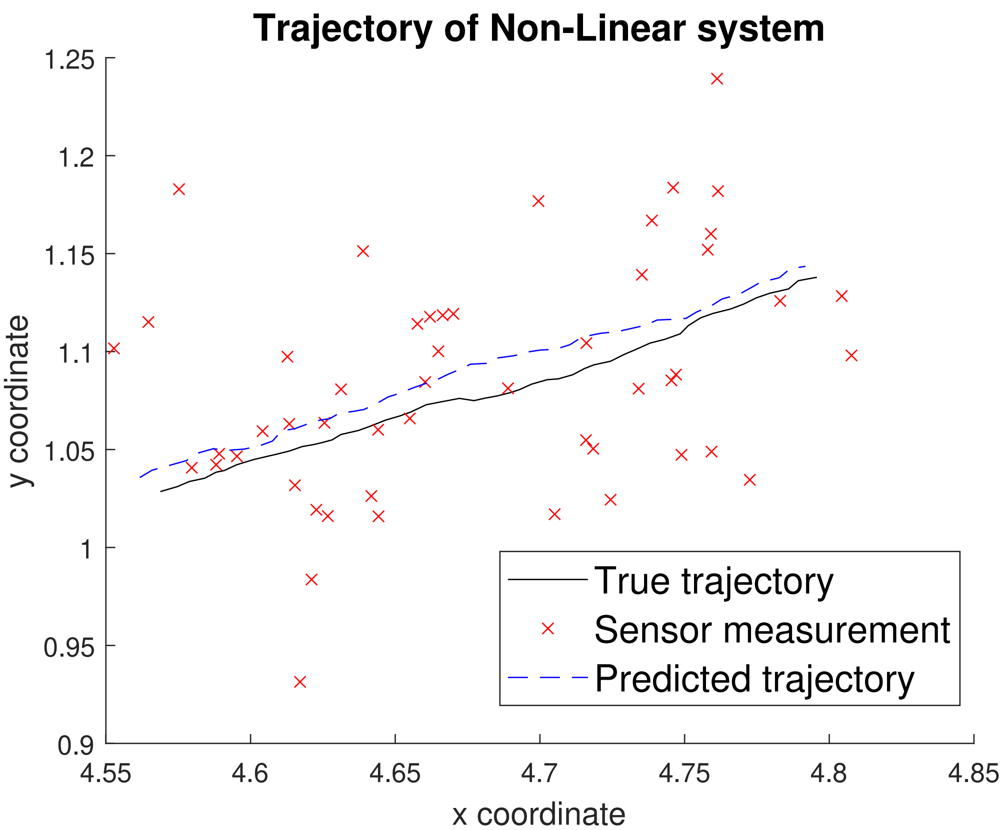

Figure 1: Trajectory of Non linear Object