

# THE EXTENDED KALMAN FILTER

## 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:

$$x_k = Fx_{k-1} + V_k \quad (1)$$

$$y_k = Hx_k + W_k \quad (2)$$

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:

$$x_k = f(x_k) + V_k \quad (3)$$

$$y_k = h(x_k) + W_k \quad (4)$$

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 linearizing the plant about the optimal value and consecutively applying the bayesian 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  $x_k$ :

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

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:

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

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:

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \\ \phi_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + T v_k \cos(\phi_{k-1}) \\ y_{k-1} + T v_k \sin(\phi_{k-1}) \\ \phi_{k-1} + T \omega_k \end{bmatrix} + \begin{bmatrix} V_{1,k} \\ V_{2,k} \\ V_{3,k} \end{bmatrix} \quad (5)$$

$$\mathbf{y}_k = \begin{bmatrix} \sqrt{x_k^2 + y_k^2} \\ \tan^{-1} \frac{y_k}{x_k} \end{bmatrix} \quad (6)$$

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

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

## Definitions

The co-variance Matrices of Noise were taken as follows:

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

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

MATLAB's  $randn()$  function was used to generate the the noise vectors.

## 0.1 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.

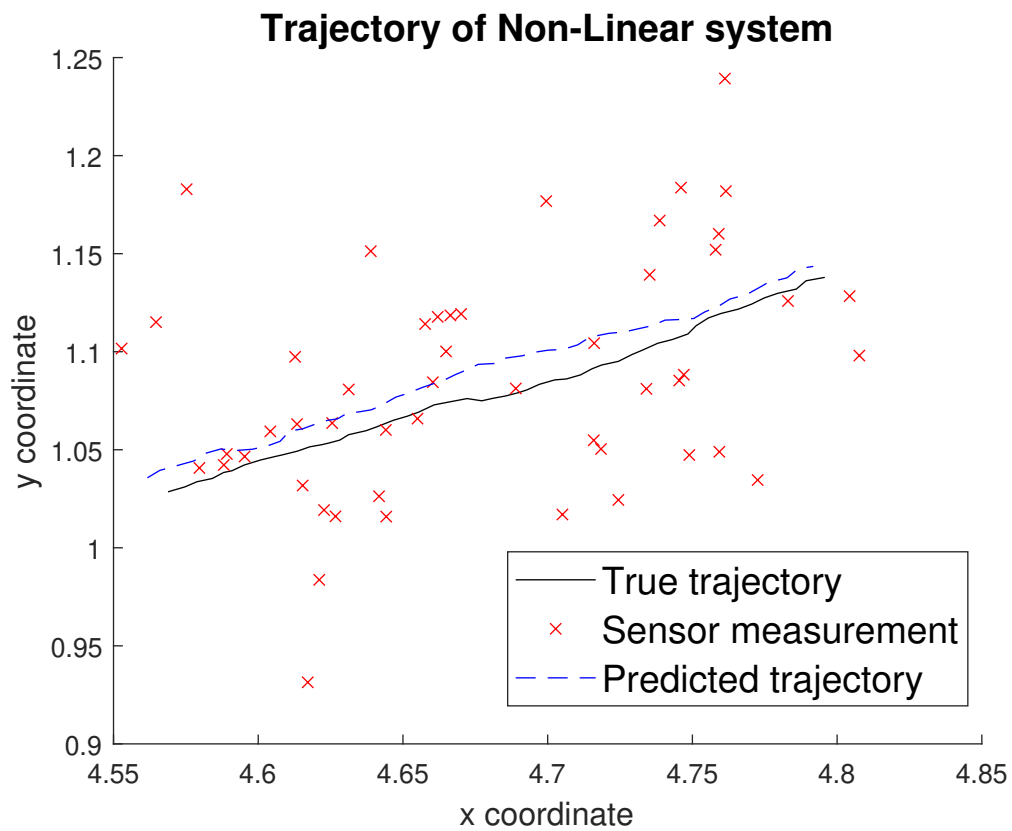


Figure 1: Trajectory of Non linear Object