Name: Manoneet Gawali Roll Number:231030035

Laboratory Exercise Report – 5

Objective

The objective of this exercise is to understand the workings of Extended Kalman Filter (EKF) and implement EKF to a simple 2D navigation problem.

Methodology

In this laboratory, you will be estimating the trajectory of a moving car (in 2D plane only). The car can observe distances from four control points (fixed). The locations of the control points and distance observations from each control point at 1 Hz are the same as in Lab-3. You are required to write programs in this exercise, and you are free to use any programming language including but not limited to C/C++, Python, MATLAB etc. Please make suitable assumptions while solving this problem. You will also be preparing a report documenting the complete details of this exercise. In the report, you should clearly explain

1. All assumptions made
2. State space model
3. Measurement model
4. Linearization steps (if any)
5. Step by step process of trajectory estimation using EKF.

Conceptualization of Problem

The Problem was given with a dataset of four Control point and its co-ordinates were given, Alongside, a car is moving in a trajectory, and it has been observed that the distance between these co-ordinates and the point on the trajectory needs to be computed. Now, distances between two points can be calculated using the Euclidian distance formula but as we know the distance between each point on trajectory and control point, the least square method for computing the co-ordinate should be used. As a result we here will first linearizing the State transition model and Measurement model in Kalman filter which we say it as Extended Kalman filter.

Approach for Problem

Now, as a given control points are given,

Control Points		
Point No.	**X**	**Y**
1	-10	0
2	0	-10
3	10	0
4	0	10

Assumptions for the problem:

- The system model considers as non - linear.
- Measurement and Processed noise assumed to as Gaussian (follow Normal - Distribution)
- State and Measurement model are assumed to be known.
- Ignore the higher order terms.
- Error is additive in nature.
- Noise and measurement noise are assumed to be uncorrelated with each other.
- Assuming body is accelerating at very slow speed, so a~0.

Functional model for independent equation, for non-linear case in LS estimate is:

$$F\ (X_a) = L_a$$

As here distances are given so we use Euclidian distance as our observation equation model,

$$d1 = \sqrt{(x1 - x)^2 + (y1 - y)^2}$$

$$d2 = \sqrt{(x2 - x)^2 + (y2 - y)^2}$$

$$d3 = \sqrt{(x3 - x)^2 + (y3 - y)^2}$$

$$d4 = \sqrt{(x4 - x)^2 + (y4 - y)^2}$$

Now, linearizing the functional model of Observation equation using Taylor's series expansion,

$L = f\,(x, y) = f\,(x0, y0) + (\, \partial d / \partial x1 * dx1\,)_0 + (\, \partial d / \partial y1 * dy1\,)_0 + (\, \partial d / \partial x2 * dx2\,)_0 +$

$(\, \partial d / \partial y2 * dy2\,)_0 +$ ....

For No-linear case we perform iteration by assuming some initial value to equation so as a result we get solution by repeating iterations until solution converges to given condition, Here in this problem I've given the value convergence limit to 10^{-15}.

Now, we proceed with our solution and compute Vectors:

X_k = predicted state vector for the next time T_k

$X_{k-1} =$ filtered state vector at time Tk

H_k = Transition matrix from time T_{k-1} to T_k , which is the Jacobian of the dynamic model with respect to the current state vector

Step 1 : Identify State Matrix Xk

$$Xk = \begin{bmatrix} x \\ y \\ Vx \\ Vy \end{bmatrix}$$

Step 2 : Measurement Vector Zk

$$Zk = \begin{bmatrix} d1 \\ d2 \\ d3 \\ d4 \end{bmatrix}$$

Step 3 : State transition model and Prediction

State transition models for the EKF of this will be,

$$X_k = X_{k-1} + (V_{k-1})_x \,.\, dt$$

$$Y_k = Y_{k-1} + (V_{k-1})_y \,.\, dt$$

$$(V_k)_x = (V_{k-1})_x$$

$$(V_k)_y = (V_{k-1})_y$$

Now from this transition model our Prediction will be

$$X_{k|k-1} = F_k \,.\, X_{k-1|k-1} + E_{X\ k-1}$$

But in this problem, we ignore $E_{X\ k-1}$, Hence our equation will be,

$$X_{k|k-1} = F_k \,.\, X_{k-1|k-1}$$

where,

$$F_k = \begin{bmatrix} 1 & dt & 0 & 0 \\ 0 & 1 & dt & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Step 4: Measurement Model

$$Z_k = H_k \,.X_k - + F(X_0) - H_k.\, X_0$$

Where,

$$H_k = \frac{\partial f}{\partial x}\Big|_{x=0}$$

Step 5: Choosing Co-variances.

$$P_{k-1|k-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Q_{k-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Step 6: Now we Initialize state vector Matrix and Weight Matrix, The initial estimate X_k and P will be:

$$Xo = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$P_o = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Step 7: Now we predict the state estimate and error covariances and make iterations which does the process of Updating the predicted state vector (Xk) to obtain the filtered state vector, as in above state we ignore a E term.

Predicted State, $X_{k|k-1} = F_k . X_{k-1|k-1}$

Step 8: Update the covariance matrix of the predicted state vector to obtain the covariance matrix of the filtered state vector:

Predicted co-variance $P_{k|k-1} = F_k . P_{k-1|k-1} . F_k^T + Q_{k-1}$

Step 9: Now Compute Gain

$$K_k = (P_k . H_k^T) [(H_k . P_k . H_k^T) + R_k]^{-1}$$

Step 10 : Compute Measurement

$$Z_k = \begin{bmatrix} dxk \\ dyk \\ Vxk \\ Vyk \end{bmatrix}$$

Step 11 : Update State vector

$$X_{k\,|\,k} = X_{k-1|\,k-1} + K_k \,.\, (Z_k - H_k.\, X_{k|k-1} - E_{Zk-1})$$

Step 12 : Update Co-variance

$$P_{k\,|\,k-1} = (I - K_k\,.\,H_k)\,.\,P_{k|k-1}$$

Result

Sr. No Lb Matrix	$\mathbf{X_0}$	$\mathbf{Y_0}$
Lb1	-9.7648	-9.7648
Lb2	-8.7526	-8.7526
Lb3	-7.7501	-7.7501
Lb4	-6.741	-6.741
Lb5	-5.7553	-5.7553
Lb6	-4.7879	-4.7879
Lb7	-3.8418	-3.8418
Lb8	-2.8927	-2.8927
Lb9	-1.9337	-1.9337
Lb10	-0.97136	-0.97136
Lb11	1.1638e-07	1.1638e-07
Lb12	0.97136	0.97136
Lb13	1.9337	1.9337
Lb14	2.8927	2.8927
Lb15	3.8418	3.8418
Lb16	4.7879	4.7879
Lb17	5.7553	5.7553
Lb18	6.741	6.741
Lb19	7.7501	7.7501
Lb20	8.7526	8.7526
Lb21	9.7648	9.7648

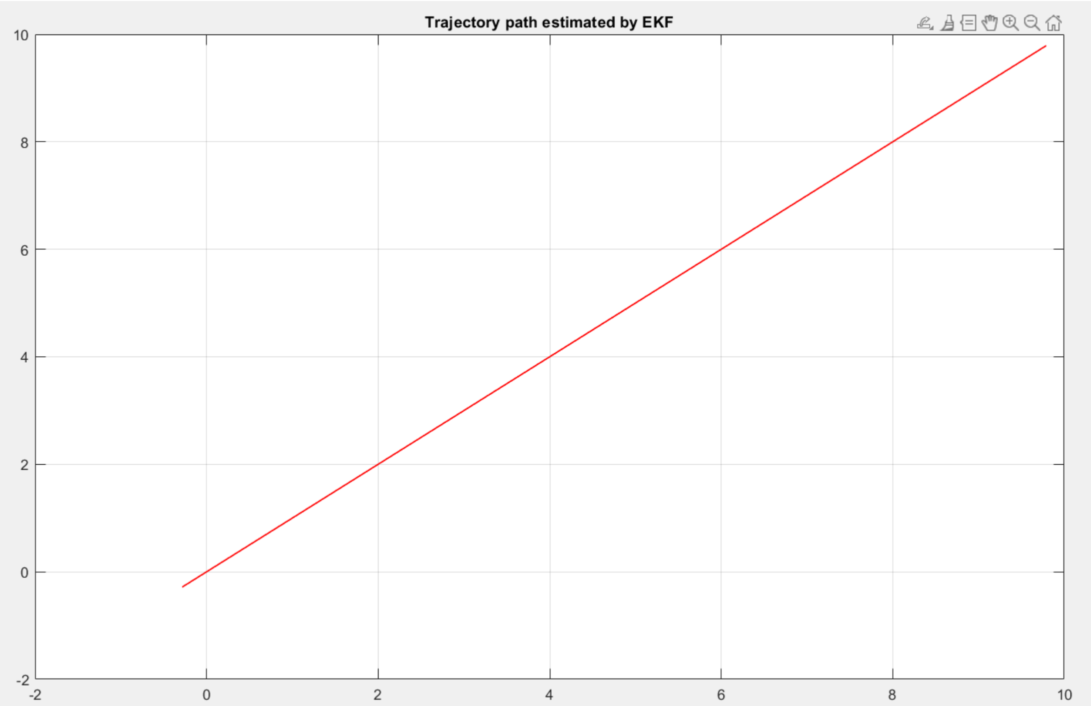


Figure 1: Trajectory Path of car estimated using EKF

Discussions

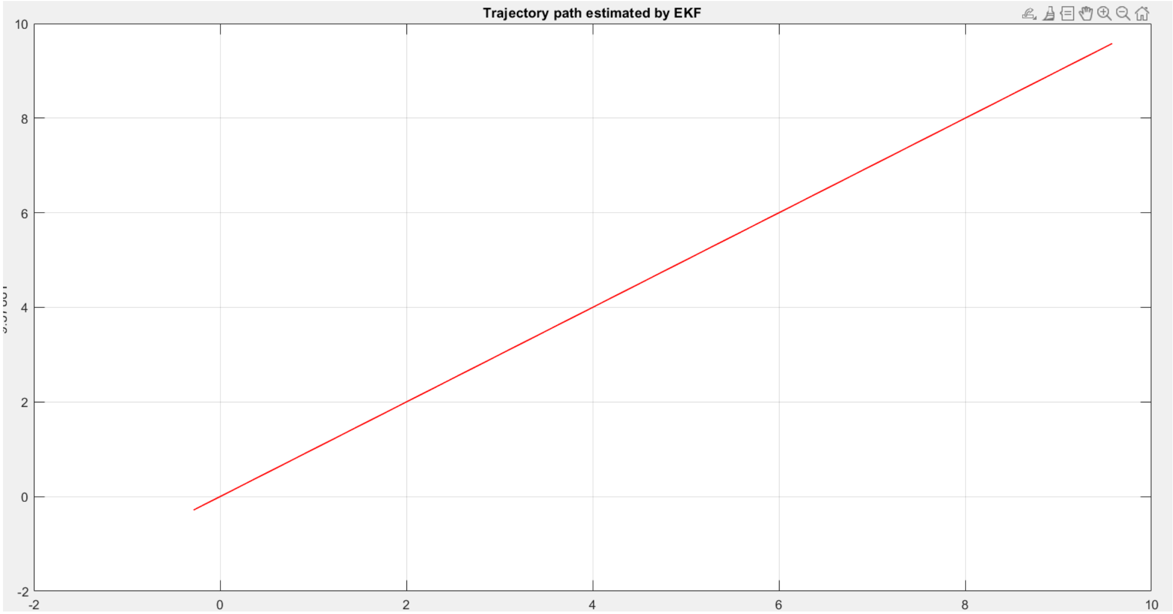
1. How did you choose the initial state and covariance?
   The Initial estimate we choose is from solving the simultaneous equations we get values of x and y, vx and vy values we took 0 as we assumed that acceleration is nearly 0 and car start from its initial state.
2. How will you go about choosing the process noise covariance and measurement noise covariance?
   Based on the understanding of system dynamics, uncertainties and variations in the process are modelled. These uncertainties can arise due to factors such as sensor noise, environmental changes, modelling simplifications, or errors in the system's mathematical representation. The uncertainties are quantified and organized into a covariance matrix. Process Noise Covariance matrix describes the covariance (or correlation) between different elements of the process noise. A larger covariance indicates a higher degree of uncertainty or variability in the corresponding processes. The process noise covariance matrix is not fixed and may need adjustments based on how well the Kalman filter performs in tracking the system. Techniques such as covariance adaptation or tuning can be used to refine the process noise covariance matrix during the filtering process.

   For Measurement of Noise covariances matrix is to understand that different sensors have different levels of uncertainty and noise in their measurements. These Sensors are often calibrated and tested to understand their behaviour and performance characteristics. This process helps in quantifying the uncertainties associated with sensor measurements. Calibration involves adjusting the sensor

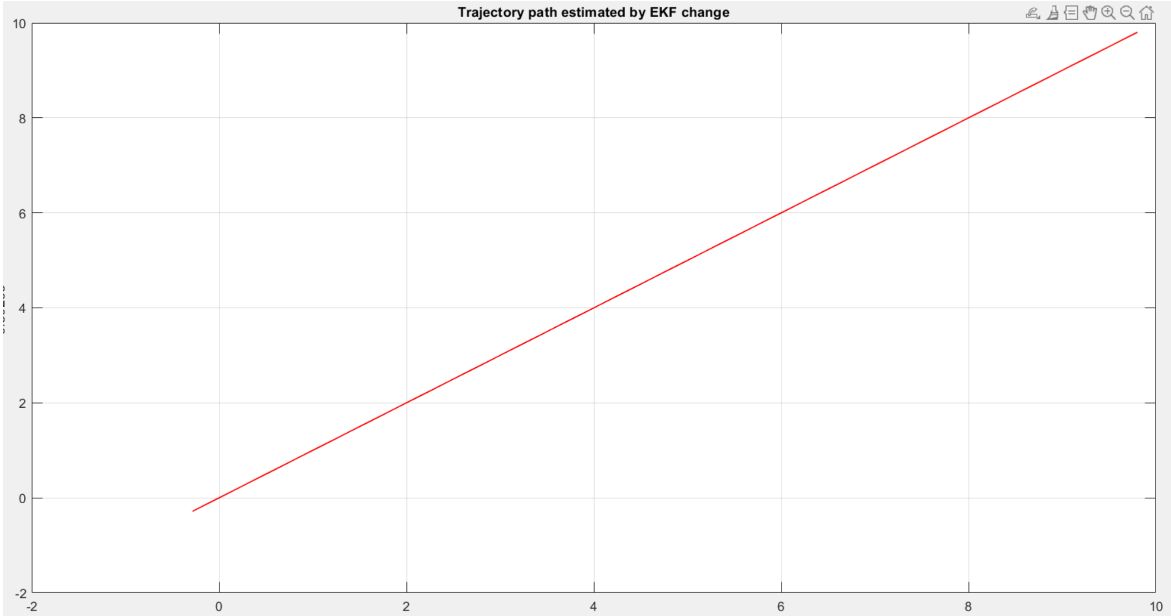
outputs to match known reference values, while testing involves evaluating sensor performance under various conditions.
In Summary we can say that the measurement noise covariance matrix is essential in Kalman filtering as it captures the uncertainties and variations in sensor measurements. By accounting for measurement noise, the Kalman filter can effectively weigh the influence of sensor data during state estimation and produce more accurate and reliable results.

3. Generate the results for different values of process noise and measurement noise covariances. Do you observe any differences? If yes, try to explain the reasons for these differences.

Q taken as diag of 0.001
R as 1



Q taken as diag of 0.01
R as 2



Points	x	y
1	-9.7646	-9.7646
2	-9.0926	-9.0926
3	-8.2380	-8.2380
4	-7.3204	-7.3204
5	-6.3910	-6.3910
6	-5.4545	-5.4545
7	-4.5048	-4.5048
8	-3.5286	-3.5286
9	-2.5305	-2.5305
10	-1.5342	-1.5342
11	-0.5507	-0.5507
12	0.4179	0.4179
13	1.3669	1.3669
14	2.3054	2.3054
15	3.2247	3.2247
16	4.1428	4.1428
17	5.0892	5.0892
18	6.0703	6.0703
19	7.1379	7.1379
20	8.2119	8.2119
21	9.2732	9.2732

4. Plot a graph of trace of posterior estimation error covariance vs. time. What can you understand from this graph?

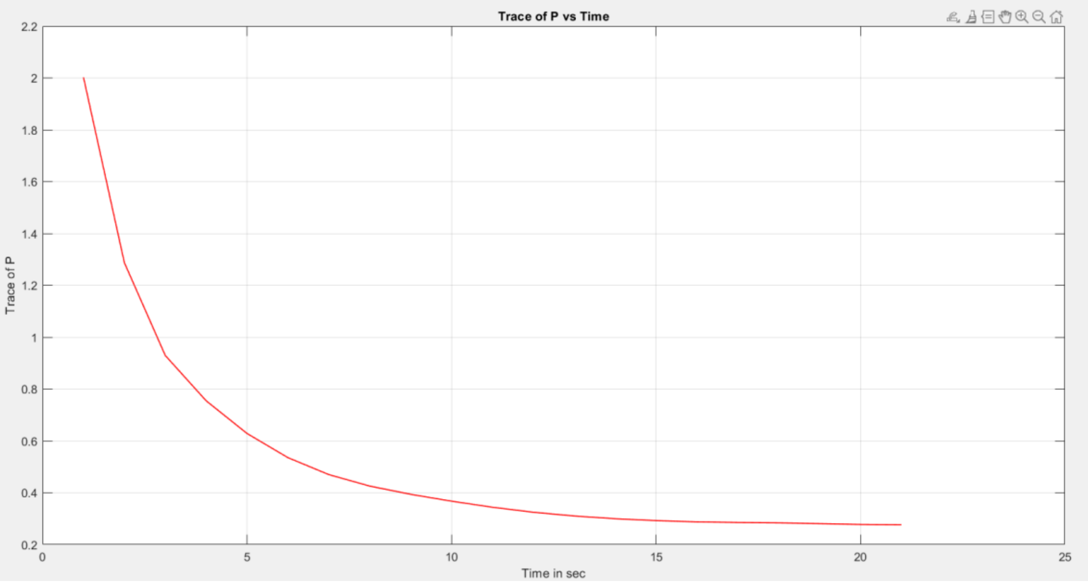


Figure 2: Trace of P vs Time.

Values of trace of P decreases with time which is observed in the graph, this shows that the confidence of filter is increasing as more measurement values are incorporated. This shows uncertainty in state is reduced .

5. Did you need to linearize either the state transition or measurement model? If yes, why? How did you choose the nominal value about which the model is linearized?
   Yes in this case we linearized the measurement model, because as the equation of Euclidian's distance is non-linear hence we linearize.
   we need to linearize the measurement model as it is important to linearize when the case is Non-Linear, so for this we linearize non-linear model by Taylor's series expansion. Linearizing around the points for discreate models are used to approximate the non-linear functions around specific points. Also, in addition if the system is linear there is no necessity to linearization step, so in this Kalman filter directly acts without linearization and generate values in such case.

Learning Outcomes

1. The Lab intended to give a clear understanding of how Extended Kalman filter model works and estimate predictions and changes according to the results obtained.
2. EKF provides the optimal estimate of the system state in the least squares sense, minimizing the mean squared error of the estimate.
3. EKF are designed to handle various types of noise in measurements, such as random noise, process noise, and measurement noise. They use probabilistic models to update the state estimate.

References

[1] D. Goel, "Lecure Notes," in *IIT Kanpur*, UP, 2024.

[2] G. A. Bartone, GNSS,Inertial Navigation and Integration, New Jersey: Wiley, USA, 2013.

[3] P. D. Groves, Principle of GNSS,inertial and Multisensor integrated Navigation Systems, Boston: Artech House, 2013.