

Object Tracking in Simulink using Extended Kalman Filter

Nivedita¹, Pooja Chawla²

M.Tech Student, Computer science and Engineering, Institute of Technology & Sciences, Bhiwani, India¹

Assistant Professor, Computer science and Engineering, Vaish P.G. College, Bhiwani, India²

Abstract: Moving object tracking has been widely used in various areas such as intelligent transportation system, video surveillance applications and airport security system and so on. In this paper, we are explaining an example of how to implement an Extended Kalman filter in Simulink. The problem of tracking flying objects is implemented using Extended Kalman Filter. Simulink is an environment for simulation and model-based design for dynamic and embedded systems. Extended Kalman Filter is the nonlinear version of Kalman Filter.

Keywords: Kalman filter, LQE, EKF, Simulink.

1. INTRODUCTION

Automatic tracking of object is difficult task for a large range of home, corporate, and industrial applications. The video cameras are the most commonly used sensors in a large number of applications which are starting from surveillance to smart rooms for video conferencing. Moving object tracking means to find different locations of the moving object in the video. There is a need to develop algorithm for task such as moving object detection.

Kalman filter, is an algorithm that uses a series of measurements observed over time, containing noise and other inaccuracies, and produces estimates of unknown variables. It is also known as linear quadratic estimation (LQE). Extended Kalman filter (EKF) linearizes about an estimate of the current mean and covariance. Simulink provides an interactive graphical environment and a customizable set of block libraries that let you design, simulate, implement, and test a variety of time-varying systems, including communications, controls, signal processing, video processing, and image processing.

2. PROBLEM FORMULATION

The EKF is simply an *ad hoc* state estimator that only approximates the optimality of Bayes' rule by linearization. One of the earliest applications of the Extended Kalman Filter was to solve the problem of tracking flying objects. The basic problem is shown in Figure 1.

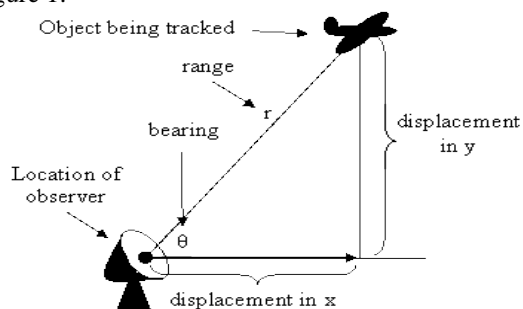


Figure 1: Relationship between displacements and range-bearing.

At each point in time the object being tracked has a given range and bearing from the observer. Often the observer is considered the location of a radar dish tracking the object. The range and bearing are generated from displacements (i.e. distances from the observer) in both the x and y directions.

The tracking problem involves estimating not only the x and y displacements of the object but also its x and y velocities. Given that the displacements and velocities are non-linearly related to the range and bearing this is an ideal problem to solve using an Extended Kalman Filter.

The Extended Kalman Filter algorithm requires the calculation of Jacobian matrices for the state and measurement equations. These take the following forms. Over a small period of time the displacement can be considered to change according to the first order approximation,

$$\hat{\mathbf{x}}(k+1) = \begin{bmatrix} \hat{x}_{pos}(k+1) \\ \hat{x}_{vel}(k+1) \\ \hat{y}_{pos}(k+1) \\ \hat{y}_{vel}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_{pos}(k) \\ \hat{x}_{vel}(k) \\ \hat{y}_{pos}(k) \\ \hat{y}_{vel}(k) \end{bmatrix} = F_k \hat{\mathbf{x}}(k)$$

The above equation says that over a small period of time the position changes by Δt times the velocity (in both the x and y directions) and that the velocity remains constant (in both the x and y directions). F_k is the required Jacobian matrix.

3. A SIMULINK IMPLEMENTATION

A Simulink model that implements the basic tracking problem discussed above and which uses an Extended Kalman Filter to estimate the object's trajectory is shown in Figure 2.

There are two main parts to this model: firstly the blocks that model the actual trajectory of the object being tracked; and secondly the Extended Kalman Filter used to estimate the trajectory of the object from measured data.

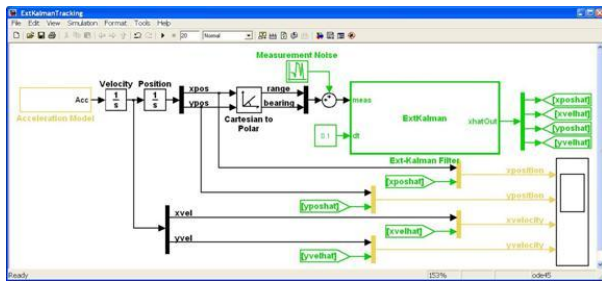


Figure 2: Simulink Model for Tracking a Flying Object using an Extended Kalman Filter.

The Actual Trajectory

The blocks that are coloured black are used to model the actual trajectory of an object flying in 2-dimensional space. Ultimately the properties being measured are the range and bearing. These are calculated from the x and y displacements, which are generated by integrating velocities, which in turn are generated by integrating accelerations.

The accelerations are generated by the acceleration model shown in Figure 3. A simpler model could use either a constant velocity or a constant acceleration. This model assumes serially correlated random accelerations. This corresponds to an object that is manoeuvring with a velocity that cannot change too quickly.

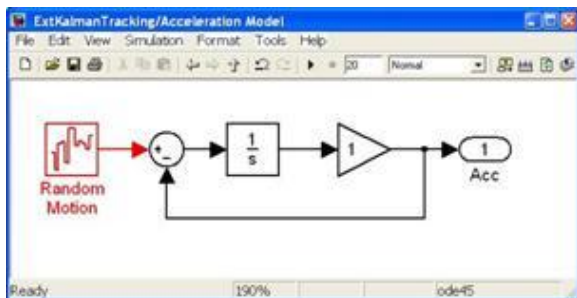


Figure 3: Acceleration Model.

In this example the object is assumed to start in the north-west and be travelling due east at 100 m/s. This corresponds to the settings given in the following table,

Quantity	Value
Initial x position	-1000 m
Initial y position	-1000 m
Initial x velocity	100 m/s
Initial y velocity	0 m/s
x displacement noise power	10
y displacement noise power	0.1

The Extended Kalman Filter

The Extended Kalman Filter uses a predictor-corrector algorithm to estimate unmeasured states of a discrete process. For the tracking problem under consideration the measured data is the object's actual range and bearing

corrupted with zero-mean Gaussian noise and sampled at 0.1s intervals. The range noise has a variance of 50 while the bearing noise has a variance of 0.005.

The Extended Kalman Filter itself has been implemented using an Embedded MATLAB Function block. The block is discrete with a sample time of 0.1 seconds. The code for the block is shown below.

```
function xhatOut = ExtKalman(meas,dt)
% This Embedded MATLAB Function implements an extended Kalman
filter used
% for object tracking.
%
% The states of the process are given by
% x = [x_position; x_velocity; y_position; y_velocity];
%
% and the measurements are given by
% y = [range; bearing]
%
% where
% range = sqrt(x_position^2 + y_position^2)
% bearing = atan2(y_position/x_position)
%
% Author: Phil Goddard (phil@goddardconsulting.ca)
% Date: Q2, 2011.
%
% Define storage for the variables that need to persist
% between time periods.
persistent P xhat Q R
if isempty(P)
% First time through the code so do some initialization
xhat = [-900;80;950;20];
P = zeros(4,4);
Q = diag([0 .1 0 .1]);
R = diag([50^2 0.005^2]);
end
% Calculate the Jacobians for the state and measurement equations
F = [1 dt 0 0; 0 1 0 0; 0 0 1 dt; 0 0 0 1];
rangeHat = sqrt(xhat(1)^2+xhat(3)^2);
bearingHat = atan2(xhat(3),xhat(1));
yhat = [rangeHat; bearingHat];
H = [cos(bearingHat) 0 sin(bearingHat) 0;
     -sin(bearingHat)/rangeHat 0 cos(bearingHat)/rangeHat 0];
% Propagate the state and covariance matrices
xhat = F*xhat;
P = F*P*F' + Q;
% Calculate the Kalman gain
K = P*H'/(H*P*H' + R);
% Calculate the measurement residual
resid = meas - yhat;
% Update the state and covariance estimates
xhat = xhat + K*resid;
P = (eye(size(K,1))-K*H)*P;
% Post the results
xhatOut = xhat;
```

Simulation Results

Simulation results are shown in Figure 4. Each of the actual and estimated states are shown on a separate axis, with the actual trajectory shown in blue and the estimated trajectory shown in red.

The filter takes some time to converge to an acceptable estimate. However after 10-15 seconds the estimates track the actual trajectories to a reasonable accuracy.

The estimates could be further improved by tuning the noise covariance estimates before (and during) applying the filtering.

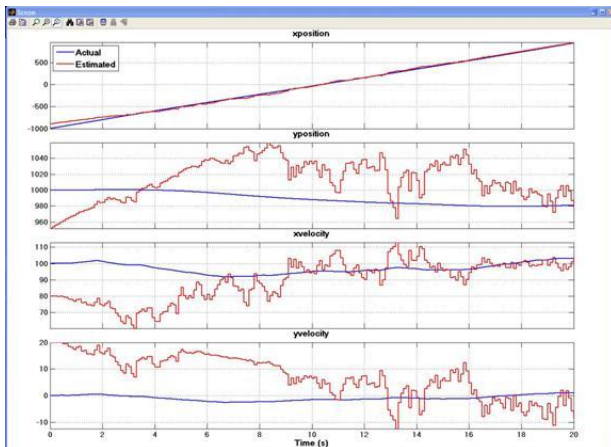


Figure 4: Actual and Estimated Positions and Velocities.

4.CONCLUSION

Simulink is the best, easy and quick method to implement the problem of tracking of flying objects with reasonable accuracy. In other words, with the help of block diagrams and graphical results, the estimated value achieve high accuracy.

REFERENCES

- [1] A. Gyaourova, C. Kamath, S. and C. Cheung - Block matching object tracking - LLNL Technical report, October 2003
- [2] C. Ridder, O. Munkelt, and H. Kirchner - Adaptive Background Estimation and Foreground Detection using Kalman-Filtering - Proceedings of International Conference on recent Advances in Mechatronics (ICRAM), pp. 193-199, 1995
- [3] E. T. Baumgartner, H. Aghazarian, and A. Trebi-Ollenu. "Rover Localization Results for the FIDO Rover." Jet Propulsion laboratory, CalTech, Pasadena, CA.
- [4] Gary Bishop and Greg Welch. "An Introduction to the Extended Kalman Filter". University of North Carolina SIGGRAPH 2001 course notes. ACM Inc., North Carolina, 2001.
- [5] John Spletzer. "Extended Kalman Filter". Lecture notes CSC398/498. Lehigh university. Bethlehem, PA, USA. March 2005.
- [6] Jonas Dino. "NASA Ames Research Center, Moffett Field, Calif., history related to the Apollo Moon Program and Lunar Prospector Mission." NASA News. Available [Online] Last modified: March 2005.
- [7] Y. Rosenberg and M. Werman – Real-Time Object Tracking from a Moving Video Camera: A software approach on PC - Applications of Computer Vision, 1998. WACV '98. Proceedings.

BIOGRAPHY

Nivedita, is now pursuing M.Tech in Computer Science and Engineering and **Pooja**, is now working as Assistant Professor in Vaish P.G. College, Haryana. Both of them are the author of many articles. Their research interests are Mobile Ad hoc Networks and Sensor Networks and are always supported by their father.