International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 07, July-2018

Design of Extended Kalman Filter for Object Position Tracking

D.S. Inaibo[1], M.Olubiwe[2], C.A.Ugoh[3], R.E.Echendu[4]
1,2,3,4, Department of Electrical and Electronic Engineering,
Federal University of Technology, Owerri, Nigeria

Abstract **- This study present the design of extended Kalman filter (EKF) for object position tracking. It is required to accurately track the position of an object amidst noisy measurements. The state variables and nonlinear output equations were obtained for a flying object at a fixed point position. An extended Kalman filter and its algorithm was developed in the embedded Matlab/Simulink function block. The measurement noise was introduced in the filter using the random noise block of the Matlab/Simulink block code. Simulations were performed at 0.1s sampling intervals. The output error standard deviation was varied between 0.05 and 1. This results to optimal selection of the system noise covariance matrix. The simulation results obtained showed that the designed extended Kalman filter accurately tracked object position with improved filtering performance by effective tuning of the noise covariance matrix.**

Key words: Extended, Object, Position, Noise, Track

I INTRODUCTION

One of the fundamental problems in control and mobile computation is the ability to determine the position of an object. The question then arises on which position of the object to track and how to locate the position of the object; what kind of system model can be used for the object motion and which type of noise is it taking, still remains to be answered. There is a complication often encountered in object tracking due to the measurement of noise. In order to predict the true position of a moving object, the noise must be filtered. There are two types of noise: the measurement noise caused by errors in tracking device and the state (or process) noise caused by perturbation due to environmental factors. Several researches have been carried out to solve this problem; research is still ongoing to improve on existing works in the literature. In order to track the exact position of an object, a Kalman filter or an extended Kalman filter (a modified version of Kalman filter) can be used.

The Kalman filter has become the main focus of research and application, especially in the field of autonomous or assisted application [1]. Kalman filter is used widely in control systems as a predictor-corrector for estimating unmeasured states of plant. Though Kalman filtering is applicable in many fields, it is mainly used for estimation and estimators' performance analysis [2]. As a set of mathematical equations, Kalman filter gives an efficient computational means of estimating the state of a process such that the mean of the squared error is minimized. As a very powerful filter that has been used in various aspects, it can be used to perform the estimation of: the past, present, and even future states, and even in situation when the exact nature of modeled system is not known [3]. The Kalman filter also called linear quadratic estimator (LQE) is an estimator for solving the linear quadratic problem. The linear quadratic problem is the problem of estimating the instantaneous condition of a linear dynamic system disturbed by white noise (or statistical noise), that is, random signals whose processes are zero. Hence, to solve the linear quadratic problem, a LQE algorithm is developed using a series of measurements observed over time containing white noise and some other errors, and then produces estimates of unknown variables that are more accurate than those obtained from a single measurement. In practice, it has helped to perform many tasks that would not have been possible without it. Its immediate areas of applications have been in the control of intricate dynamic systems like: in continuous manufacturing processes, aircraft, ships, or spacecraft. In order to control dynamic system in control engineering, it is expected that the variables of process to be controlled be known or measurable. In the applications mentioned earlier, it may not be possible or desirable [2] to have all the variables required to be controlled measured. In this case, the Kalman filter helps to make available a means for deducing the missing information from indirect (and noisy) measurements. A major limitation to the Kalman filter is largely due to the fact that it requires both the dynamic system and the measurement functions to be linear in respect to the state variables [4]. In most of the practical applications, these requirements are not likely satisfied. This has led to the development of a modified version of the Kalman filter to what is now known as extended Kalman filter (EKF), which can be applied in a situation when the system and/or the measurement models are not linear. As the name implies, the extended Kalman filter (EKF) is an extension to the standard Kalman filter that linearises a system beyond the linear system limitation of the Kalman filter to nonlinear system. The EKF works exactly as a Kalman filter when used on linear system because all the linearization performed on the linear system would give the same result as in the application of the standard Kalman filter. One of the benefits of using the extended Kalman filter is that it provides simple approach to solving nonlinear systems compared to other approaches which employs Kalman filter in solving nonlinear systems. Hence in using EKF to solve nonlinear problem, a code is written which is more transparent and less complex or simplified algorithm is developed. With a less complex algorithm for filtering, the execution time of EKF will be shorter than the more complex versions of the Kalman filter for nonlinear systems. The extended Kalman filter (EKF) being extension of Kalman filter is a state estimator which optimally approximates Bayesian rule used in Kalman filter by linearization. In one of its applications, the EKF is used to solve the problem of tracking flying objects. An object is tracked at each point in time as well as its given range and bearing from the observer. The tracking problem involves estimating the horizontal and the vertical displacements and position of the object as well as the velocities.

II OBJECT TRACKING

Tracking can be referred to as the inference shape of object, appearance, and motion with respect to time. Object tracking is the case where the position of another object is to be estimated in terms of measurements of relative range (or distance) and angles to one's position [5]. [6], defined tracking as a problem which involves inference generation about a motion of an object given a sequence of images. For object tracking to be performed properly, the most likely measured potential target location should be used to update the object's state estimator. This is known as the data associated problem [1]. The possibility of the obtained measurement being accurate lies between the predicted states of the object and the measured state. In object tracking applications, the most famous techniques for object's updating is to assume that the dynamics of the object can be modeled, and that noise affecting the object dynamics and feedback sensor data is stationary.

The Kalman filter and its extension, the extended Kalman filter, is a location based approach for finding object locations in the next frame [7]. The object center is first found, and then uses the filter to predict the position of it in the next frame. In the case of a Kalman filter, it is used to estimate the state of a linear system where the state is assumed to be distributed by a Gaussian. If the system is nonlinear, the extended Kalman filter is used so as to linearize the system. It consists of two steps: prediction and correction as shown in Fig 1. It provides an optimal estimation of a moving object at each step time which contains some kind of white noise, and some noisy observations about its position. A typical problem of a flying object whose position is being tracked is shown in Fig 2.

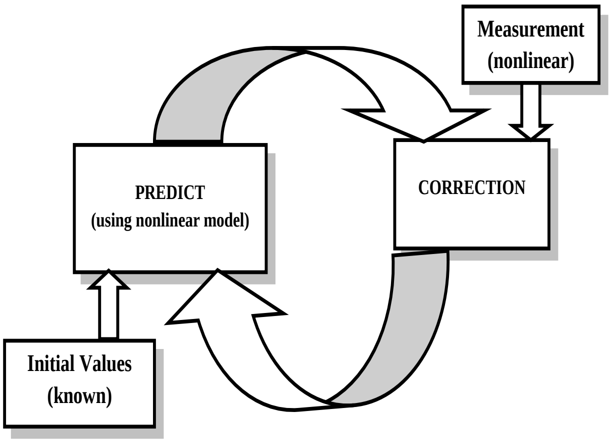

Fig1: EKF predict/correct model

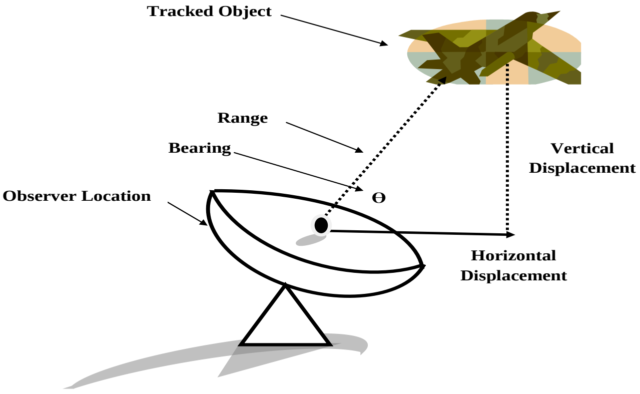

Fig. 2: A typical problem of object whose position is being tracked

Object tracking is relevant in the following areas:

- In predicting stock prices of the stock market which is fully random and nonlinear
- In motion-base recognition such as human identification based on gait, and object detection
- In automated surveillance, that is, to monitor a scene so as to detect activities or events which are suspicious.
- In human to computer interaction, that is, a form gesture recognition, eye gaze tracking for data input to computers
- In traffic monitoring to enhance traffic flow management.
- In automation annotation and retrieval of the videos in multimedia and database

Some types of object tracking are defined below:

- Extended object tracking: In this type of object tracking, an object generates multiple measurements per time and measurements are spatially structured around the objects, that is, a case where an object occupies multiple resolution cells.
- Group object tracking: In this case, a group objects which consists of two or more sub-objects that share certain motion in common. Also, the objects are not tracked separately rather are treated as a unit. Hence, the group objects occupy many resolution cells; and each sub-object is likely to occupy either one or more resolution cells. It should be noted that each object generates multiple measurements per unit time and measurements are spatially arranged around the objects.
- Tracking with multi-path propagation: Each object in this case creates multiple measurements per unit time as a result of the multi-path propagation but measurements are not spatially structured around object.
- Point object tracking: each objects creates at most a single measurement per unit time, that is, a single resolution cell is occupied by an object.

III LINEARIZATION

Linearization is a linear approximation of a nonlinear system about a valid region (operating point). For nonlinear dynamic systems, linearization is done by computing the Jacobian matrix of the state space equation or in most cases use of Simulink control design software.

Consider the continuous time nonlinear differential function as shown below

$\dot{x} = f(x(t), u(t), \Delta t)$ (1)

$y(t) = g(x(t), u(t), \Delta t)$ (2)

Where:

$x(t)$ *represents the system states*

$u(t)$ *represents the inputs to the system*

$y(t)$ *represents the outputs of the system*

The linearized model of the system defined by the equations above is valid around the region defined by the operating conditions, which in most cases are given by the initial conditions (states),

$\Delta x(t) = x(t) - x_o$ (3)

$\Delta u(t) = u(t) - u_o$ (4)

$\Delta y(t) = y(t) - y_o$ (5)

The linearized models in terms of the defined new variables are represented by:

$\Delta\dot{x}(t) = A\Delta x(t) + B\Delta u(t)$ (6)

$\Delta y(t) = C\Delta x(t) + D\Delta u(t)$(7)

Where A, B, C and D are time varying matrices.

Linearization finds use in systems model analysis and control design applications. Also software (Simulink control design) are used to linearize continuous time, discrete time models which results in linear time variant model in state space form [8]. Simulink software linearizes models using a block by block approach. This approach linearizes each block in the Simulink model and combines the results to produce the linearization of the specified system. Nonlinear dynamic systems describing changes in variables with respect to time are normally unpredictable as such linearization of such a system is needed for further analysis of the system state. Typically the behavior of nonlinear system is described by sets of nonlinear equations with unknown variable or functions represented in polynomial form of degree higher than one and they are normally difficult to solve, hence nonlinear systems are commonly linearized.

IV CONTINUOUS AND DISCRETE TIME COMPARISON

Though the majority of the Kalman filters are implemented in discrete time, there exists a continuous time counterpart. The discrete time implementation is largely due to the fact that the filters are implemented in digital computers. The continuous time implementation is not practical in many cases. However, that does not mean that modeling of systems in continuous time cannot be handled with discrete time Kalman filter. An example is demonstrated to compare both forms.

Consider a system in continuous time form as follows:

$$\dot{x}(t) = F_c(t)x(t) + G_c(t)u(t) + w(t) \quad (8)$$

Where F$_c$ and G$_c$ indicate the system matrices for a continuous system, and w$_c$is the continuous time process noise vector with covariance matrix Q. The continuous model can be discretized, e.g. using a first order approximation, as in

$$x_k = x_{k-1} + \Delta t\,\dot{x}_k \quad (9)$$

Where Δt is the sampling time, or the duration of a single discrete time step, thus resulting in $t = k\Delta t$.

Then, including the continuous time model gives

$$x_k = x_{k-1} + \Delta t\left[F_c(t)x(t) + G_c(t)u(t) + w(t)\right] \quad (10)$$

Which can be simplified to obtain the following discrete time system

$$x_k = \left[I + \Delta t F_c(k\Delta t)\right]x_{k-1} + \Delta t G_c(k\Delta t)u_k + \Delta t w_c(k\Delta t) \quad (11)$$

V THE DESIGN

The design approach in this work is to assume a radar antenna (observer) monitoring the position of an aircraft (object being tracked) in an airport at a certain height above ground level. The kinematic of the entire tracking system is obtained under the following assumptions.

a A given range and elevation angle (or bearing) is assigned to the object being tracked at each point in time from the observer.

b At a given period of time, the object position changes by Δt times the velocity in both the horizontal (x) direction and the vertical (y) direction.

c The velocity remains the same in both the x and y directions.

d The effect of gravitational force, air drag, and rotation motion are ignored.

In order to realize the objective of this research, the assumptions stated above are maintained and are used to derive the kinematic equations in this section. The state variables of the position of an aircraft at certain ground level are obtained, first in continuous time, and later transformed to its equivalent discrete time state. An extended Kalman filter (EKF) equation is definedand an algorithm for EKF established in Matlab/Simulink.

In this work, acceleration which serves as the vector input, is generated by the acceleration model developed using the Matlab/ Simulink Code as shown in Fig 3.

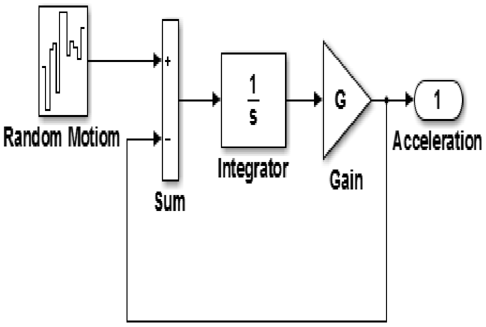

Fig 3: Input vector (acceleration model)

In this work, the model is assumed to have serially correlated random accelerations. This is the same as an object maneuvering with a velocity that cannot change quickly. The rate at which the velocity is allowed to change is determined by the gain in the first order filter.

An essential area of applying extended Kalman filter is the creation of the model using mathematical knowledge to derive nonlinear transition function for unknown parameter in each estimation state. Two models are available for use with an extended Kalman filter. These are

A The State Model:

$$x_k = f\left(x_{k-1}, u_{k-1}\right) + w_{k-1} \quad (12)$$

where x_{k-1} represents the state model which comprises parameters used for each state prediction, x_k represents the next (or new) state whose predicted data (or information) is to be obtained from the use of the transition function which is nonlinear function, u_{k-1} represents the control input which is optional (because an extended Kalman filter can be designed without a controller as in this work), and W_{k-1} is the Gaussian white noise. The variable f is the vector value nonlinear state transition function, which describes how the state is generated from t-1 without control or noise

B The Measurement Model:

$$y_k = h\left(x_k\right) + v_k \quad (13)$$

Where y_k denote the measurement model, v_k denote the Gaussian white noise affecting the output, and h is the vector-value, nonlinear measurement (or output) function, which describes how to map the state X_k to an observation Y_k

In order to take care of the nonlinear functions f and h, Jacobian matrices are calculated so as to linearize the equations at each time step about the previous state as in equations (14) and (15).

The Jacobian matrix, is defined as:

$$F_{k-1} = \left.\frac{\partial f}{\partial x}\right|_{\mathbf{x}_{k-1}} \quad (14)$$

The Jacobian matrix H_k is calculated using:

$$H_k = \frac{\partial h}{\partial x} = \begin{bmatrix} \frac{\partial r_k}{\partial x_k} & \frac{\partial r_k}{\partial y_k} \\ \frac{\partial \theta_k}{\partial x_k} & \frac{\partial \theta_k}{\partial y_k} \end{bmatrix} = \begin{bmatrix} \frac{x_k}{\sqrt{x_k{}^2 + y_k{}^2}} & \frac{y_k}{\sqrt{x_k{}^2 + y_k{}^2}} \\ -\frac{y_k}{\sqrt{x_k{}^2 + y_k{}^2}} & \frac{x_k}{\sqrt{x_k{}^2 + y_k{}^2}} \end{bmatrix} \quad (15)$$

Therefore, $$H_k = \begin{bmatrix} \cos\theta_k & \sin\theta_k \\ -\sin\theta_k & \cos\theta_k \end{bmatrix} \quad (16)$$

Hence, applying the calculated Jacobian matrices, an extended Kalman filter (EKF) algorithm can be developed. Note: W_{k-i} and $V_k$are random variables and are independent and normally distributed with covariance R and Q

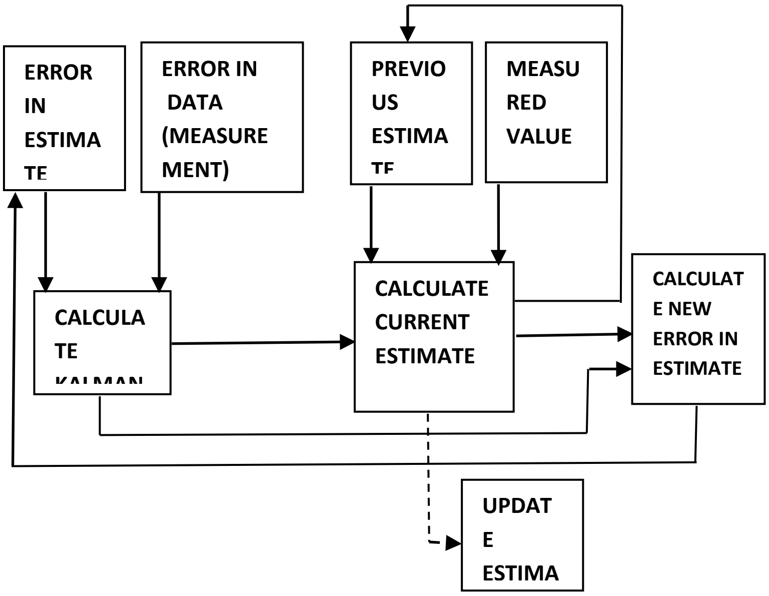
ERROR IN ESTIMATE
ERROR IN DATA (MEASUREMENT)
PREVIOUS ESTIMATE
MEASURED VALUE
CALCULATE KALMAN
CALCULATE CURRENT ESTIMATE
CALCULATE NEW ERROR IN ESTIMATE
UPDATE ESTIMA

FIG 4 Block diagram model for EKF Algorithm

A Kalman Filter Gain

One of the main elements of the Kalman filter is the gain,K. An understanding of the Kalman gain will give an insight on how much of the measurement to use to update the new estimated or predicted state.

Fig 5 below shows the quantities that go into the gain K, which are the error in estimate and error in measurement. It should be noted that error in estimate represents the projected (or estimated) error covariance while the error in measurement represents update (measurement) error covariance. The term covariance is used when estimation and measurement are performed in two or more dimensions. As established below for the purpose of the design in this work, the gain is the ratio of estimate error to sum of the estimate error and measurement error and is taken to be a value between zero (0) and one (1) as expressed mathematically below

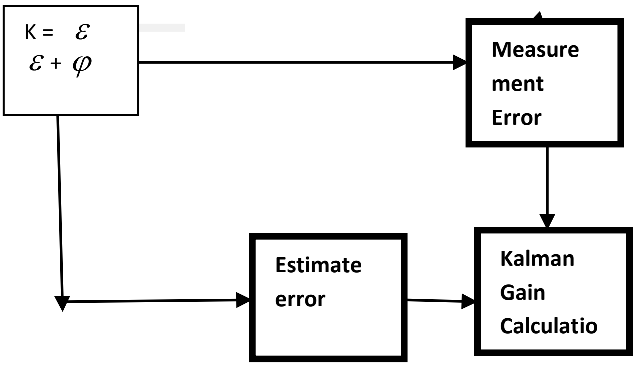

Fig 5: A typical block diagram for Kalman gain Calculation

The relationship between the gain, estimate error, measurement error, new or current estimate, previous estimate, error in current estimate, error in previous estimate, and measurement is established mathematically below.

$$\mathbf{K} = \frac{\varepsilon}{\varepsilon + \varphi};\ 0 \le \mathrm{K} \le 1 \qquad (17)$$

Where ε = estimate error, and φ = measurement error.

$$E_t = E_{t-1} + \mathbf{K}\left(\mathrm{M} - E_{t-1}\right) \quad (18)$$

Where:

E_t = current estimate

E_{t-1} = previous estimate

M = measurement.

$$\varepsilon_t = \frac{\varphi\varepsilon_{t-1}}{\varphi + \varepsilon_{t-1}} \qquad (19)$$

Which implies that

$$\varepsilon_t = \left(1 - \mathbf{K}\right)\varepsilon_{t-1} \qquad (20)$$

Where:

ε_t = current estimate error

ε_{t-1} = previous estimate error

Fig 6, shows the pattern of relationship between the gain, the measurement and the estimate.

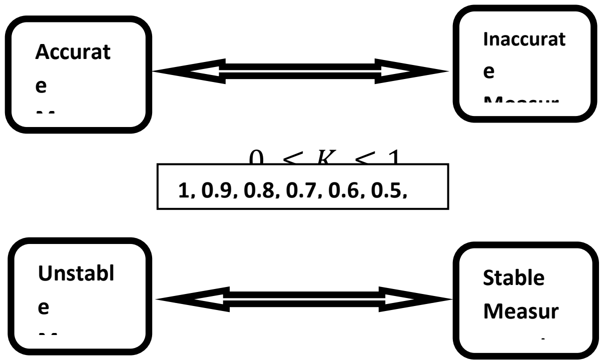

Fig 6: Relationship pattern between gain, measurement and estimate

As shown in Fig 6, if the gain is high or large that is close to 1, it indicates fairly accurate measurement with unstable estimate. The idea is to ensure that the difference between measurement and previous estimate as in Equation (17) is so small so that the value obtained by multiplying it with the gain is small. Also with high measurement error, the gain is reduced so that it is possible to have better estimates. In fact, with large gain the estimated error increases and this can make the object estimated position to be off target (see Equation 16).

In order to design a kalman filter algorithm for the purpose of tracking a flying object at a certain

height from ground level in two-dimension Cartesian plane, the recursive algorithm process is summarized as shown in Fig 7 below

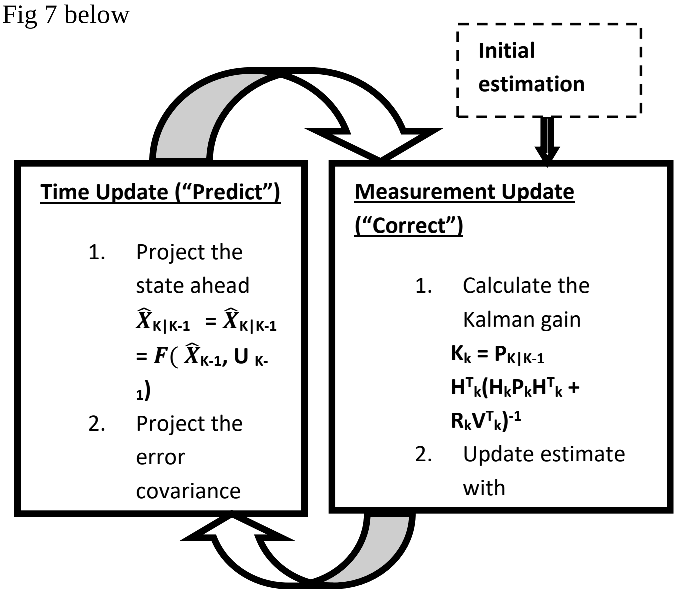
Initial
estimation
Time Update ("Predict")
1. Project the state ahead
$\hat{X}_{K|K-1} = \hat{X}_{K|K-1} = F(\hat{X}_{K-1}, U_{K-1})$
2. Project the error covariance
Measurement Update ("Correct")
1. Calculate the Kalman gain
$K_k = P_{K|K-1} H^T_k(H_kP_kH^T_k + R_kV^T_k)^{-1}$
2. Update estimate with

Fig 7: A summary of the designed EKF Algorithm process

Two main parts is associated with the design of EKF algorithm. This include: prediction state and correction state.

In the correction state, the estimation state $\hat{\mathrm{X}}_{k|k\text{-}1}$ and $\mathrm{P}_{k|k-1}$ obtained from the prediction state is used to calculate the Kalman gain or K which stands as the "trustable" value of state model and measurement variable. $\hat{x}_{k|k\text{-}1}$ is calculated using state transition model, while $P_{k|k-1}$ is user define. In any case, when the covariance matrix R_k tends to zero, it shows that measurement is more trustworthy and reliable than state model. Since $P_{k|k-1}$ is a function of the covariance matrix Q_k , it means that as Q_k tends to zero (the same as $P_{k|k-1}$ approaching zero), the measurement becomes less trustworthy. After which the estimated data is updated depending on the values of the gain matrix, K .The final stage is the updating of the error covariance for next iteration.

B Extended Kalman Filter Model

A block diagram of the developed model that implements the required object position tracking problem is presented as shown in Fig 8. It makes use of an extended Kalman filter (EKF) to estimate an object's position.

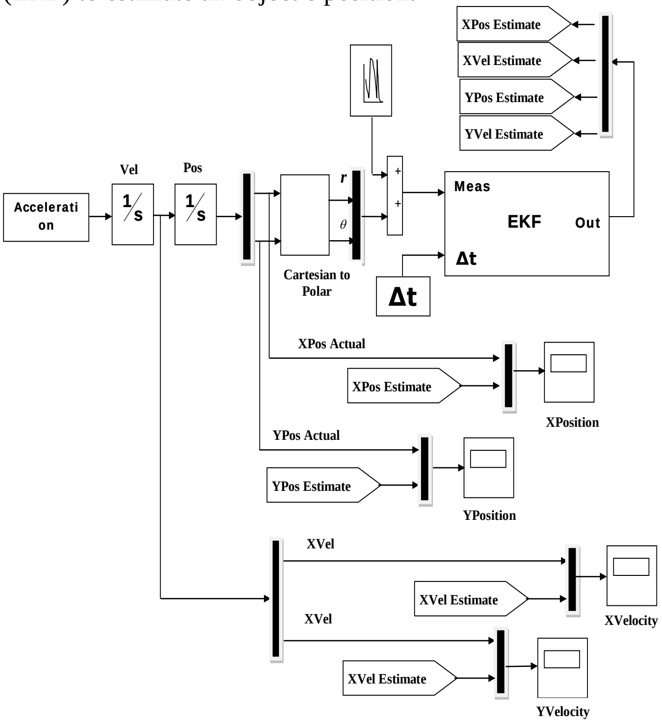

Fig 8: Block diagram model of designed EKF

In order to perform a flying object position tracking problem, the measured data is actual range *r* of the object and elevation angle θ which is corrupted with zero-mean Gaussian noise, implemented using Matlab , and at 0.1s sampled intervals, Δt denote an execution over 10 times steps. A variance of 50 was selected for the range noise while variance of 0.005 was assigned to the elevation angle noise.

The EKF is implemented using an embedded Matlab function block as shown in Fig 7. It is a discrete Matlab code block with sampled interval of 0.1s.

C Definition of Quantities

The necessary quantities used for the extended Kalman filter in this work are defined in Table 3.1. The values of the quantities used for simulations performed in this context were obtained through mathematical calculations and selection process to represent the actual states of the system as well as the noisy measurement. This is not obtainable in real-life application problem as actual (or true) states are not known rather are obtained from the measurement system.

Table 1: Parameters used and definition (Nivedita and Pooja 2015)

Parameter	Symbol	Definition
Assumed initial state error covariance matrix	P_k	$\begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$
Measurement noise covariance matrix	R_k	$\begin{bmatrix} 50 & 0 \\ 0 & 0.005 \end{bmatrix}$
Actual initial state vector	X_o	$\begin{bmatrix} 1000 \\ 1000 \end{bmatrix}$ m
Time increment	Δt	$0.1s$
Actual Initial Velocity	(v_x, v_y)	(100,0) m/s
Horizontal and Vertical noise power	-	10, 0.1

VI RESULTS AND DISCUSSION

Simulations are performed, considering a simulation step of 10 and observation error standard deviation varied between 0.05 and 1 which results to optimally selection or turning of the system noise covariance matrix. The error variance σ^2 was optimized with respect to the varied values of the error standard deviation. The simulation results for the optimized error variance selected for an object position tracking system using extended Kalman filter (EKF) presented in this work are shown in Figures 9 to 12. The essence of the simulations was to ascertain the optimality of the designed EKF.

In this work an object has been assumed to be at an actual state of 1000m for the position in both dimensions and actual state of 100m/s for the velocity for horizontal dimension and 0m/s for vertical dimension. The horizontal and vertical noise power are 10 and 0.1 as shown in Table 1.

A Optimized Error Variance of 0.05

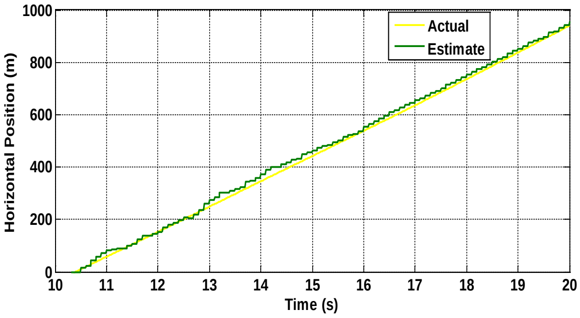

Fig.9: Estimation of object horizontal position σ^2 =0.05

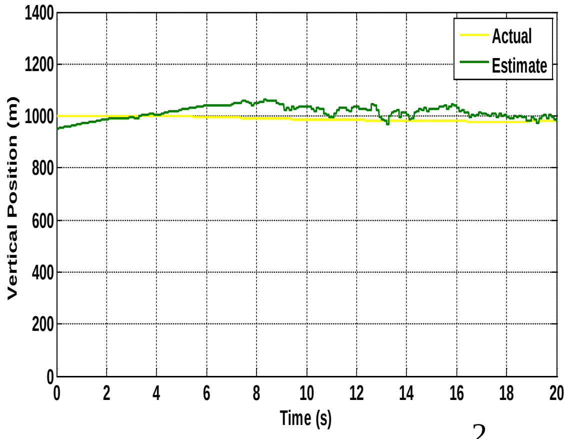

Fig. 10: Estimation of object vertical position σ^2 = 0.05

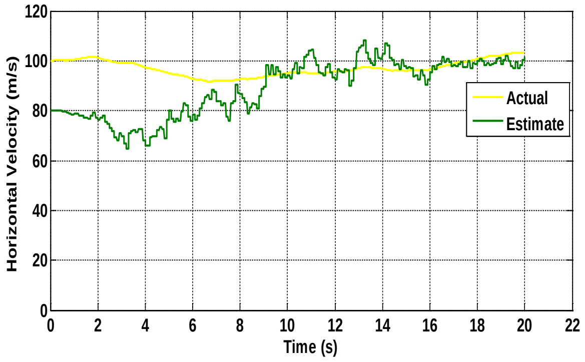

Fig. 11: Estimation of object horizontal velocity σ^2 = 0.05

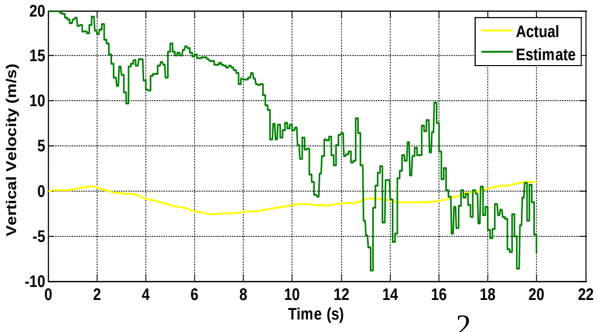

Fig. 12: Estimation of object vertical velocity σ^2 = 0.05

B Optimized Error Variance of 0.1

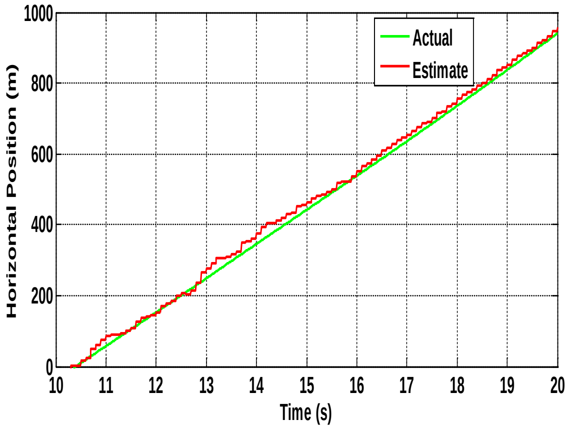

Fig. 13: Estimation of object horizontal position σ^2 = 0.1

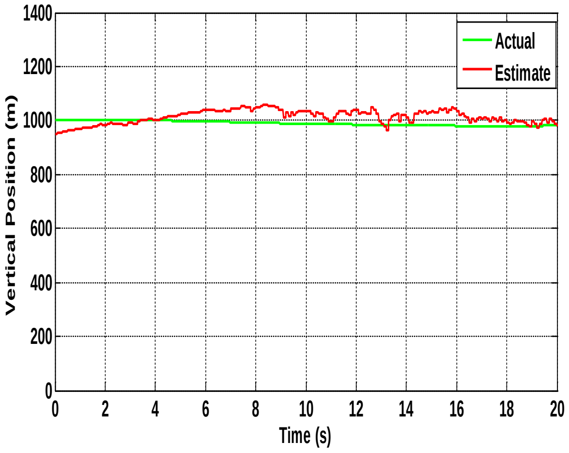

Fig. 14: Estimation of object vertical position σ^2 = 0.1

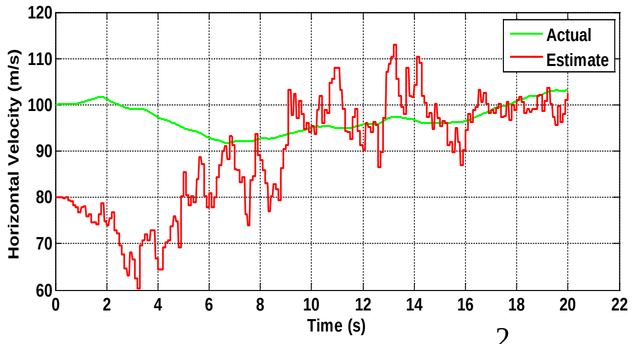

Fig.15: Estimation of object horizontal velocity σ^2 = 0.1

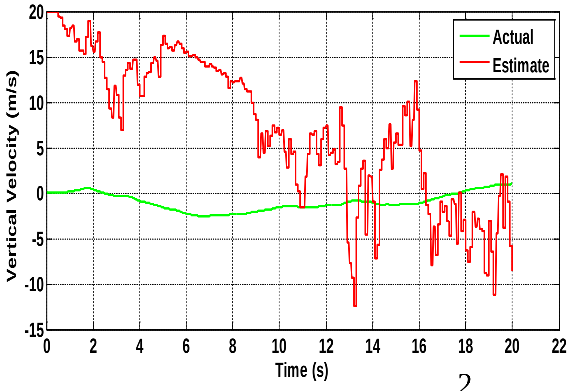

Fig.16 Estimation of object vertical velocity σ^2 = 0.1

C Optimized Error Variance of 1

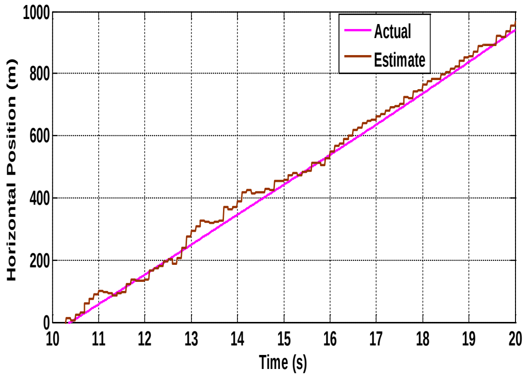

Fig. 17: Estimation of object horizontal position σ^2 = 1

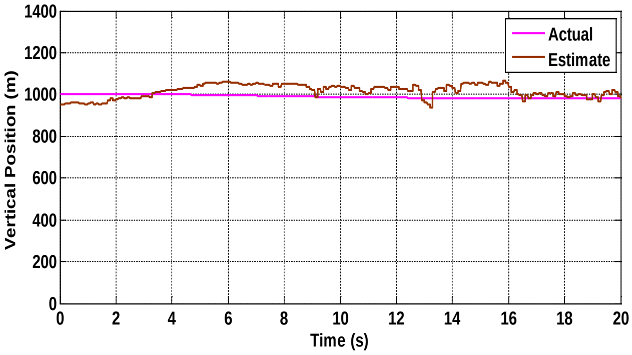

Fig. 18: Estimation of object vertical position $\sigma^2 = 1$

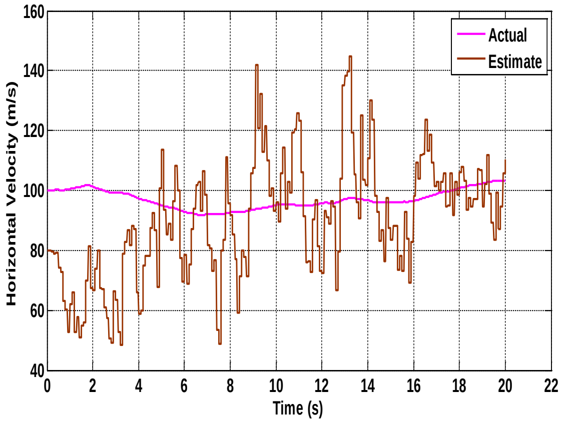

Fig. 19: Estimation of object horizontal velocity $\sigma^2 = 1$

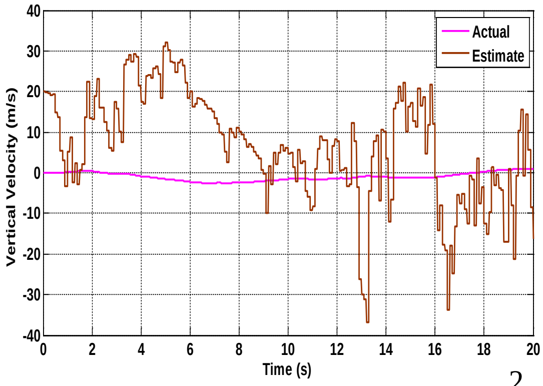

Fig. 20: Estimation of object vertical velocity $\sigma^2 = 1$

For the purpose of simulation of the system with the selected values for the optimized error variance for the system noise covariance matrix, the dynamic state of the system and the designed extended Kalman filter (EKF) were analyzed in Simulink. Figs 9-12, shows the simulation graphs of an object position tracking system using EKF when implemented with an optimized error variance of 0.05. Fig. 9 shows the horizontal position of an object at actual state of 1000m and is tracked or estimated by the measurement noise at about 10seconds. In the vertical coordinate as shown in Fig.10, the object position 1000m is approximately estimated by the measurement noise at about 11-20 seconds. In Fig.11, the actual state of the object velocity at 100m/s is estimated at about 9.5-12seconds and a better estimate at 16-20seconds. Consequently, in Fig.12, an actual state of object velocity was estimated at about 14- 20 seconds. The Figures below shows the simulation graphs for the implemented EKF for an object position tracking system with an optimized error variance of 0.1. In Fig 13 shows the horizontal position of an object at actual state of 1000m and is tracked or estimated by the measurement noise at about 10seconds. The actual state of object at vertical position is approximately estimated by the measurement noise at about 11-20 seconds as shown in Fig.14. In Fig.15, the actual state of object velocity in the horizontal dimension is approximately tracked by the estimate at about 16-19seconds. The graph of Fig.16 shows that actual state velocity of object is tracked by the estimate at 14-20seconds.

The simulation results for an object position tracking system using EKF is implemented in this work with an optimized error variance of 1, is shown in Figures below. It can be seen that for simulation result in Fig.17, the actual state of object position in the horizontal is tracked at about 10 seconds. In the vertical coordinate as shown in Fig.18, the actual object position 1000m is approximately estimated by the measurement noise at about 13-20 seconds. Consequently, in Fig.19, the actual object velocity is tracked by estimated time of 17- 19 seconds and in Fig.20 the actual object velocity is initially tracked at about 1 second and later at about 9-14seconds.

Though the convergence of the extended Kalman filter (EKF) takes some time, it can be seen that the tracking performance of the EKF is optimally improved within the selected error variance chosen in this work. The estimate was able to track the actual state (or trajectory) of an object to an appreciable accuracy after 10-15seconds.

VII CONCLUSION

This work has presented design of an extended Kalman filter for object position tracking. The estimation of the states of an object position in two dimensions has been studied using noisy measurement. In order to realize the objective of this work, the kinematic equations representing the state of an object at a particular position above ground level were formulated in two dimensional coordinates. The formulation assumed a radar antenna (observer) tracking an aircraft (object) some distance from the observer location in two dimensions. The equations were initially expressed in continuous time and were later transformed into an equivalent discrete time form using a step time of 0.1s. The values of the parameters used were obtained from existing literature. The white noise used in this work is a random noise block of the Matlab Simulink. In order to perform simulations in Matlab/Simulink environment, the kinematic equations in discrete time form were represented by their equivalent simulink blocks. This was integrated with the designed extended Kalman filter in which the Matlab code for the algorithm is embedded. The simulated results at a fixed error variance of the system noise covariance matrix obtained by optimally varying the observation error standard deviation which shows that the designed filter takes some time to converge to an acceptable estimate and track the actual states to a reasonable accuracy. Hence, a system that be used to estimate object position such as an aircraft position (or trajectory), has been implemented.

VIII REFERENCES

[1] K Rameshbabu, J Swarnadurga , G Archana, and K Menaka, ‘‘Target Tracking System Using Kalman Filter’’. International Journal of Advanced Engineering Research and Studies E-ISSN2249–8974. IJAERS/Vol. II/ Issue I, 90-94, 2012.

[2] S Mohinder, Grewal and P Angus, Andrews, ‘‘ Kalman Filtering: Theory and Practice Using Matlab’’, Second Edition, John Wiley and Sons, Inc. ISBNs: 0-471-39254-5 (Hardback); 0-471-26638-8 (electronic). pp 1-410, 2001.

[3] A Seekircher, S Abeyruwan and U Visser, ‘‘Accurate Ball Tracking with Extended Kalman Filters as a Prerequisite for a High-level Behaviour with Reinforcement Learning’’, The 6th Workshop on Humanoid Soccer robots, pp 1-6, 2011.

[4] L ASCORTI, ‘‘An Application of the Extended Kalman Filter to the Attitude Control of a Quadrotor’’, Politecnico Di MilanonCorso di LaureaMagistrale in Computer EngeneeringDipartimento di Elettronica, Informazione e Bioingegneria. pp 1- 108, 2013

[5] F Gustafsson, F Gunnarsson, N Bergman, U Forssell, J Jansson, R Karlsson and P J Nordlund, ‘‘ Particle Filters for Positioning, Navigation and Tracking’’ Final version for IEEE Transactions on Signal Processing, 2015.

[6] S Harsha and C V S Kumar, ‘‘A Critical Appraisal on Engineering Kalman-Filters for Real-Time Object Tracking & Motion Detection Systems,’’ IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 5. ISSN 2348 – 7968, 2014.

[7] S Shantaiya, K Verma and K Mehta, ‘‘Multiple Object Tracking using Kalman Filter and Optical Flow,’’ European Journal of Advances in Engineering and Technology, 2(2): 34-39. ISSN: 2394 - 658X, 2015.

[8] Nivedita and P Chawla, ‘‘Object Tracking in Simulink using Extended Kalman Filter’’, International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 7, pp 365-366, July, 2015