Ninth LACCEI Latin American and Caribbean Conference (LACCEI'2011), Engineering for a Smart Planet, Innovation, Information Technology and Computational Tools for Sustainable Development, August 3-5, 2011, Medellín, Colombia.

Multi-Sensor Data Fusion for Trajectory Prediction of a Missile

Richard Herrera Rey
University of the Llanos, Villavicencio, Meta, Colombia, rirey41@hotmail.com

Anngy Herrera Rey
University of the Llanos, Villavicencio, Meta, Colombia, annrey41@hotmail.com

William Snyder Herrera Rey
University of the Llanos, Villavicencio, Meta, Colombia, wlrey21@hotmail.com

ABSTRACT

This article presents a method of Fuzzy Logic Based Sensor Data Fusion for target-tracking applications, where the tracking is done by fusing the measurements from two independent sensors; the data from each sensor are processed by a respective Kalman Filter to estimate the states of target, the filters used were the extended Kalman filter (EKF) and Fuzzy extended Kalman filter (FEKF), which is based on fuzzy logic theory and Kalman filter. We compared the performances of the Kalman filter, fuzzy Kalman filter and Fuzzy Logic Based Sensor Data Fusion for position estimation under different circumstances. It has been demonstrated that the fuzzy Kalman filter gives better results than the EKF and the data fusion has an intermediate performance between the FEKF and EKF since they are the only ones which are managed.

Keywords: Data fusion, fuzzy logic, Kalman filter, target-tracking.

1. INTRODUCTION

The estimation is the procedure of determining the state of a system from noisy measurements, taking account of measurement errors and system disturbances. Here it was used the estimation method developed by Kalman (1960). One of the most substantial characteristics of the Kalman filter is its recursive procedure that processes measurements to obtain the optimal estimate.

For the estimation the original data from each sensor are processed by a respective Kalman Filter to estimate the states of a target (position, velocity, and acceleration). In general, a sensor measurement system will have one or more of the following problems

- An individual sensor covers only a restricted region of the environment and provides measurement data of only local events.
- A breakdown of a sensor or sensor channel, causing a loss of data from the viewed object or scene.
- Measurements from sensors depend on the accuracy and precision of the basic sensing element used in the sensor and the data gathered would thus have limited accuracy.

One of the effective solutions to the preceding problems is multisensor data fusion. The term sensor fusion means the combination of sensory data or data derived from sensory data, such that the resulting information is better than it would be if these sensors were used individually; in other words, data fusion from multiple sensors, which could be of the same type or of different types. Data fusion means combining information from several sources, in a sensible way, in order to estimate or predict some aspect of an observed scene. However, sensor or data fusion should not be considered as a universal method. Data fusion is useful if the data provided are of reasonably good quality. Just manipulating many bad data would not produce any great results it might produce some results, but at a very high cost (Wilfried, 2002). Many fused very poor quality sensors would not make up for a few good ones, and it may not be easy to fix the errors in the initial processing of the data at a later stage.

The Fuzzy Logic assists in modeling conditions that are inherently imprecisely defined, in the form of approximate reasoning, which provide decision support and expert systems with good reasoning capabilities (this is called an FL-type 1 system). The concept of Fuzzy Logic is extended to state-vector level data fusion for similar sensors and it can also be used for tuning Kalman Filter. Algorithms can be developed by considering a combination of Fuzzy Logic and Kalman Filter (Kashyap and Raol, 2008). The proper combination of Fuzzy Logic- and Kalman Filter based approaches can be used to obtain improved accuracy and performance in tracking systems. An early survey on the subject can be found in (Nirmala et al., 2005).

In this study is presented a model for predicting the missile trajectory using the extended Kalman filter (EKF), Fuzzy extended Kalman filter (FEKF) and Fuzzy Logic Based Sensor Data Fusion.

2. PROBLEM STATEMENT AND PRELIMINARIES

2.1 NOTATION

The following notation will be used in the rest of the paper.

1) x_k is true state vector.
2) F_k is state transition matrix.
3) $h(x_k)$ is the observation matrix.
4) $\hat{x}_k^-$ is the estimate of x_k before processing the measurement at time k.
5) $\hat{x}_k^+$ is the estimate of x_k after processing the measurement at time k.
6) P_k^+ denotes the covariance of the estimation error of $\hat{x}_k^+$.
7) P_k^- denotes the covariance of the estimation error of $\hat{x}_k^-$.
8) K_k denotes the Kalman filter gain.
9) T is the discretization step.

2.2 PROBLEM FORMULATION

The target-tracking model is described in Cartesian coordinate system with additive noise:

$$x_{k+1} = F_k x_k + w_k \tag{1}$$

Here,

- The vector $x_k \in \mathbb{R}^d$ consists of the position and velocity of the target $x = [x_1, x_2, x_3, x_4]^T = \left[x, v_x, y, v_y\right]$,
- The process noise $w_k \in \mathbb{R}^d$ is assumed to be white, and the zero mean with covariance matrix $Q_k \triangleq \sigma_k^2$,
- F_k is given as

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

The ground-based radar provides provides measurements of range (r_m), azimuth (θ_m). The measurement model this is given as:

$$\begin{bmatrix} r_m \\ \theta_m \end{bmatrix} = h(x_k) + v_k = \begin{bmatrix} \sqrt{x_1^{\,2} + x_3^{\,2}} \\ \tan^{-1}(\frac{x_3}{x_1}) \end{bmatrix} + \begin{bmatrix} v_r \\ v_\theta \end{bmatrix} \quad (2)$$

Here, v_r, v_θ are mutually uncorrelated and zero-mean white Gaussian noises with variances σ_r^2, σ_θ^2, respectively. In order to obtain the estimates of the trajectory it's employed the filters outlined below

2.2.1 EXTENDED KALMAN FILTER (EKF)

In this method, the measurements used for updating the states are the range, azimuth, and elevation in polar frame. The measurements are nonlinear functions of the states yielding a mixed coordinates filter. In the EKF, the initial covariance depends on the initial converted measurements and the gains depend on the accuracy of the subsequent linearization. A simple way to handle the nonlinearities (Blackman, S. 1986) is to process the radar measurements sequentially in the order of elevation, azimuth, and range, while linearizing the nonlinear equations with respect to the estimated states.

However, the extended Kalman filter can be difficult to tune and often gives unreliable estimates if the system nonlinearities are severe. This is because the extended Kalman filter relies on linearization to propagate the mean and covariance of the state (Dan Simon, 2006). Despite of this the extended Kalman filter (EKF) is the most widely applied state estimation algorithm for nonlinear systems.

The extended Kalman filter can be used as illustrated below, which is executed recursively:

Alg 1. Extended kalman filter algorithm

1. Initialize the filter as follows $\hat{x}_0^+ = E(x_0^+)$ and $P_0^+ = E\left[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T\right]$
2. For k = 1 , 2 , ..., do the following.
    a. Perform the following time update equations

$$P_k^- = F_{k-1}P_{k-1}^+F_{k-1}^T + L_{k-1}Q_{k-1}L_{k-1}^T$$
$$\hat{x}_k^- = f_{k-1}(\hat{x}_{k+1}^+, u_{k-1}, 0)$$

where the partial derivative matrices F_{k-1} and L_{k-1} are defined as follows:

$$F_{k-1} = \left.\frac{\delta f_{k-1}}{\delta x}\right|_{\hat{x}_{k-1}^+}$$
$$L_{k-1} = \left.\frac{\delta f_{k-1}}{\delta w}\right|_{\hat{x}_{k-1}^+}$$

    b. Perform the following measurement update equations

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + M_k R_k M_k^T)^{-1}$$
$$\hat{x}_k^+ = \hat{x}_k^- + K_k(y_k - h_k(\hat{x}_k^-, 0))$$
$$P_k^+ = (I - K_k H_k) P_k^-$$

where the partial derivative matrices H_k and M_k are defined as follows:

$$H_k = \left.\frac{\delta h_k}{\delta x}\right|_{\hat{x}_k^-}$$

$$M_k = \left.\frac{\delta h_k}{\delta v}\right|_{\hat{x}_k^-}$$

Since complete information of the event is usually not available, the difference exists between the true model and the used model, such that the model error cannot be avoided. It is inevitable that an untrue model will downgrade the filter performance, and model divergence may arise.

2.2.2 FUZZY EXTEND KALMAN FILTER (FEKF)

The proper combination of Fuzzy Logic- and Kalman filter-based approaches can be used to obtain improved accuracy and performance in target-tracking applications, In such systems, Fuzzy Logic can aid soft decision making in the filtering process by using fuzzy *if…then* rules for making a judgment on the use of, for example, residuals in navigating the prediction or for filtering in the direction of achieving accurate results in either tracking process, feature selection, detection, matching (Jitendra R. Raol, 2009a). The equations for the Fuzzy Kalman filter are the same as those for EKF in Alg 1 except for the following equation (Klein, 2004)

$$\hat{x}_k^+ = \hat{x}_k^- + K_k B_k \tag{3}$$

Here, B_k is regarded as an output of the Fuzzy Logic-based process variable (FLPV) and is generally a nonlinear function of the innovations ***e*** of the Kalman filter. The FLPV vector consists of the modified innovation sequence for x and y axes:

$$B_k = [b_x(b_x(k+1)b_y(k+1))] \tag{4}$$

The FLPV vector for the *x* direction is developed and then generalized to include *y* direction. This vector consists of two inputs, e_x and $\dot{e}_x$, and single output $b_x(k+1)$, where $\dot{e}_x$ is computed by (Jitendra R. Raol, 2009a) :

$$\dot{e}_x = \{e_x(k+1) - e_x(k)\} / T \tag{5}$$

Here, ***T*** is the sampling interval in seconds. The rules for the inference in FIS are generally created based on the experience and intuition of the domain expert. One such rule being (Raol and J. Singh, 2009)

$$IF\ e_x\ is\ LP\ AND\ \dot{e}_x\ is\ LP\ THEN\ b_x\ is\ LP$$

2.2.3 FUZZY LOGIC–BASED SENSOR DATA FUSION

In Kalman filter fuzzification (KFF), the original data from each sensor are processed by a respective Kalman FIlter to estimate the states of a target (position, velocity, and acceleration). The error signal for each channel is generated by taking the difference of the measured and estimated positions of the target for that particular channel. The average estimation error is computed by (Jitendra R. Raol, 2009b):

$$\bar{e}_{idn}^{FKF} = \frac{e_{X_{idn}}^{FKF}(k) + e_{y_{idn}}^{FKF}(k) + e_{z_{idn}}^{FKF}(k)}{M} \tag{6}$$

M is the total number of measurement channels, and ***idn*** is the sensor identity number. The error signals are generated by

$$\begin{aligned} e_{X_{idn}}^{FKF}(k) &= x_{m_{idn}}(k) - \hat{x}_{m_{idn}}^{FKF}(k) \\ e_{y_{idn}}^{FKF}(k) &= y_{m_{idn}}(k) - \hat{y}_{m_{idn}}^{FKF}(k) \\ e_{z_{idn}}^{FKF}(k) &= z_{m_{idn}}(k) - \hat{z}_{m_{idn}}^{FKF}(k) \end{aligned} \tag{7}$$

Where $x_{m_{idn}}(k)$, $y_{m_{idn}}(k)$ and $z_{m_{idn}}(k)$, are the target position measurements in the x, y, and z axes and $\hat{x}_{m_{idn}}^{FKF}(k)$, $\hat{y}_{m_{idn}}^{FKF}(k)$, and $\hat{z}_{m_{idn}}^{FKF}(k)$ are the corresponding estimated positions from the Kalman Filter. The fused states are given by (Nirmala et al., 2005):

$$\widehat{X}_f^{FKF} = \widehat{X}_1^{FKF} + w_1(k)(w_1(k) + w_2(k))^{-1}(\widehat{X}_2^{FKF} - \widehat{X}_1^{FKF}) \tag{8}$$

Where $\{w_1(k), w_2(k)\}$ are the weights generated by the FIS for sensor 1 and sensor 2, and the normalized values of the error signals $\bar{e}_1$ and $\bar{e}_2$ are the inputs to the FIS (associated with each sensor).

The use of the data fusion should achieve improved accuracy (reduce the uncertainty of predicting the state or declaring the identity of the observed object) and more specific inferences than could be achieved using a single sensor alone.

3. SIMULATION AND RESULTS

3.1 EXAMPLE 1

Before starting the estimation, the true initial conditions of the state vector are $x_0 = [0, 0, 0, 100]^T$. It is assumed that initial values of the state vector for kalman filters are $\hat{x}_0^+ = [0.001, 0.01, 0.001, 400]^T$ and covariance of the estimation error P_0^+ for Kalman filters is:

$$P_0^+ = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Here we assume complete confidence in the initial conditions, which are incorrect with respect to the true values.

The process noise covariance Q_k is

$$Q_k = \begin{bmatrix} 0.0001 & 0 & 0 & 0 \\ 0 & 0.005 & 0 & 0 \\ 0 & 0 & 0.0001 & 0 \\ 0 & 0 & 0 & 0.005 \end{bmatrix}$$

and the measurement noise covariance R_k is set to 900 and 0.0001 for each measurement y_k. The estimated trajectories are shown in the Figure 1 in rectangular coordinates:

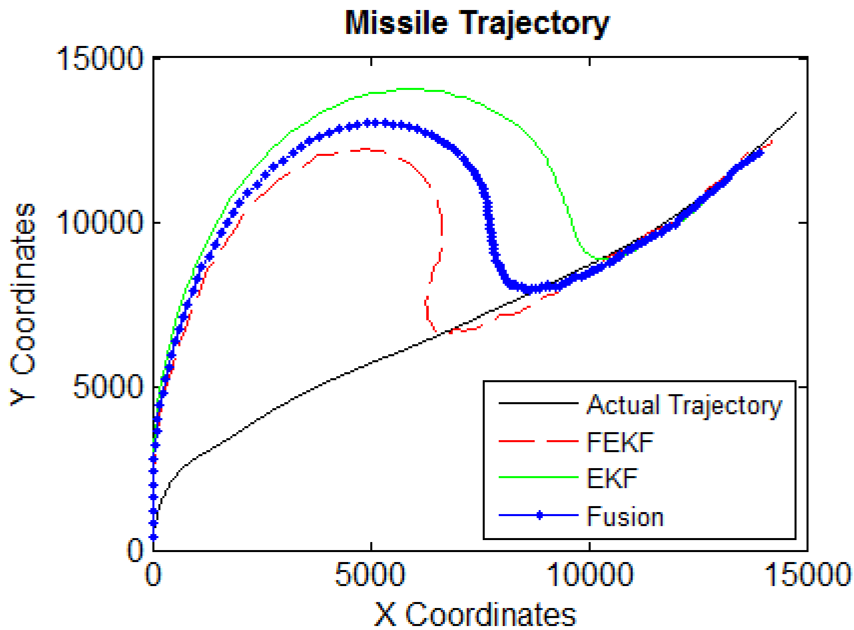

Figure 1. Comparison of actual trajectory against estimated trajectories by Kalman filters and Data fusion (X vs.Y)

The root sum square error in position (RSSPE) is shown in the Figure 2, it produces a sequence of numbers, and these values will be zero when the corresponding actual and estimated positions are exactly alike. In reality, these values will increase when the corresponding estimated positions deviate from the actual positions.

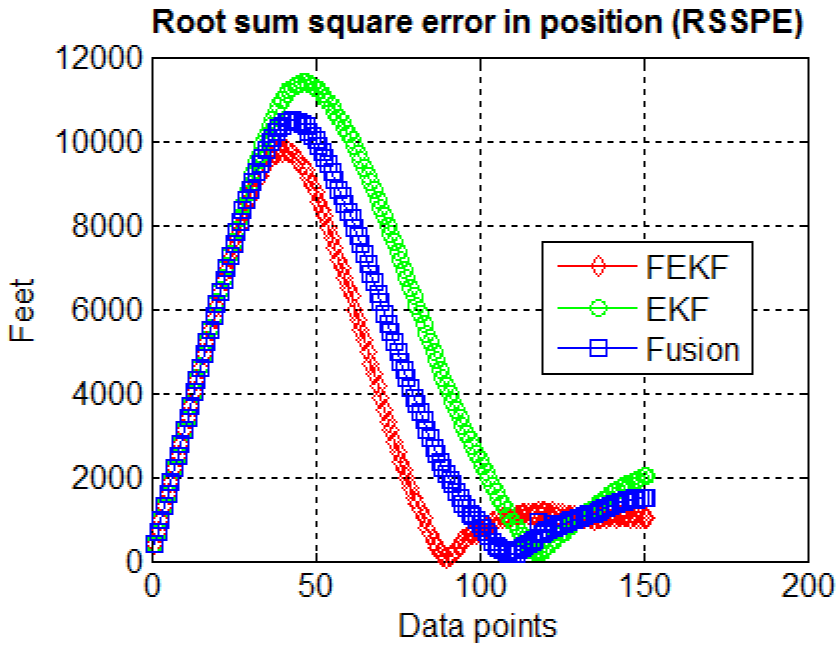

Figure 2. Root sum square error in position of the estimated trajectories by Kalman filters and Data fusion

From the Figure 1, it can be concluded that initially the prediction is inaccurate with respect to the actual trajectory because Kalman filters assume incorrect initial values. However as noisy measurements y_k are processed the predictions become more accurate, and as it's shown in Figure 2 the RSSPE decreases, moreover it can be seen that the FEKF has the best performance.

The following table summarizes the highlights from the previous example; it shows the Percentage fit error (PFE) in position of each filter

Table 1: PFE in Position

Filters	X-position	Y-position
FEKF	25.1678	63.9620
EKF	35.8717	82.3933
Data Fusion	29.1973	72.2965

This error will be zero when both actual and estimated positions are exactly alike, and it will increase when the estimated positions deviate from the actual positions.

It can be seen in the Table 1 that the FEKF gives a more accurate estimate than the EKF because it uses the Fuzzy Logic to aid soft decision making in the filtering process by using fuzzy *if...then* rules, achieving accurate results in the tracking process. Moreover it can be concluded the data fusion has an intermediate performance between the FEKF and EKF, since they are the only ones which are managed. The errors in the Table 1 are high due to the initial conditions of the filters are wrong with respect to the true value.

3.2 EXAMPLE 2

In the second execution, it is assumed the same initial values of Example 1, exept for the value of covariance of the estimation error, which is

$$P_0^+ = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}$$

Here we assume less confidence in the initial position and velocity in the Y coordinate. The estimated trajectories are shown in the Figure 3 in rectangular coordinates

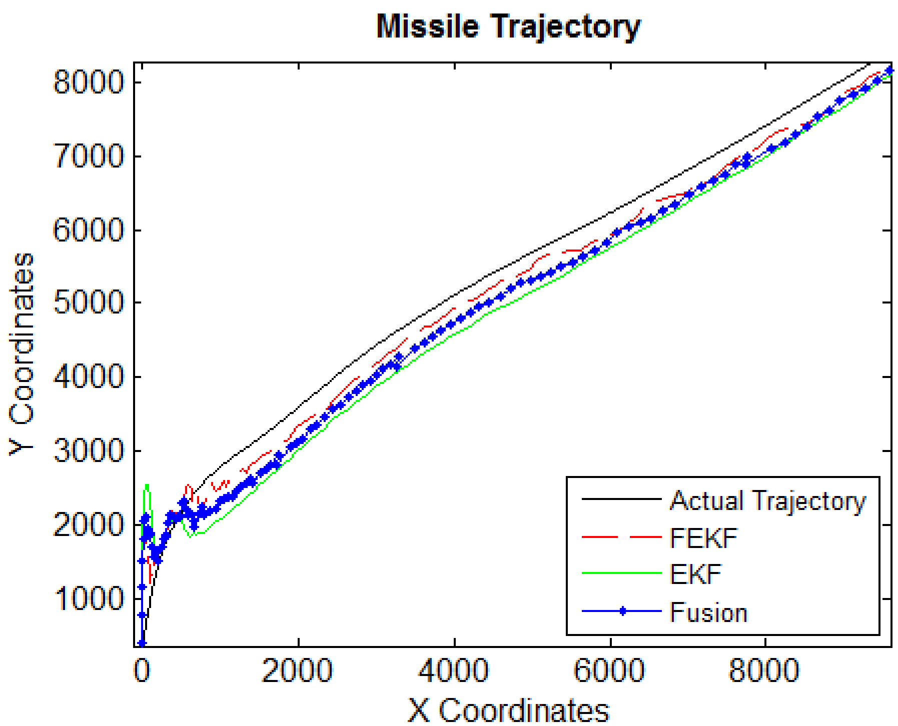

Figure 3. Comparison of actual trajectory against estimated trajectories by Kalman filters and Data fusion (X vs. Y).

The root sum square error in position (RSSPE) is shown in the Figure 4,

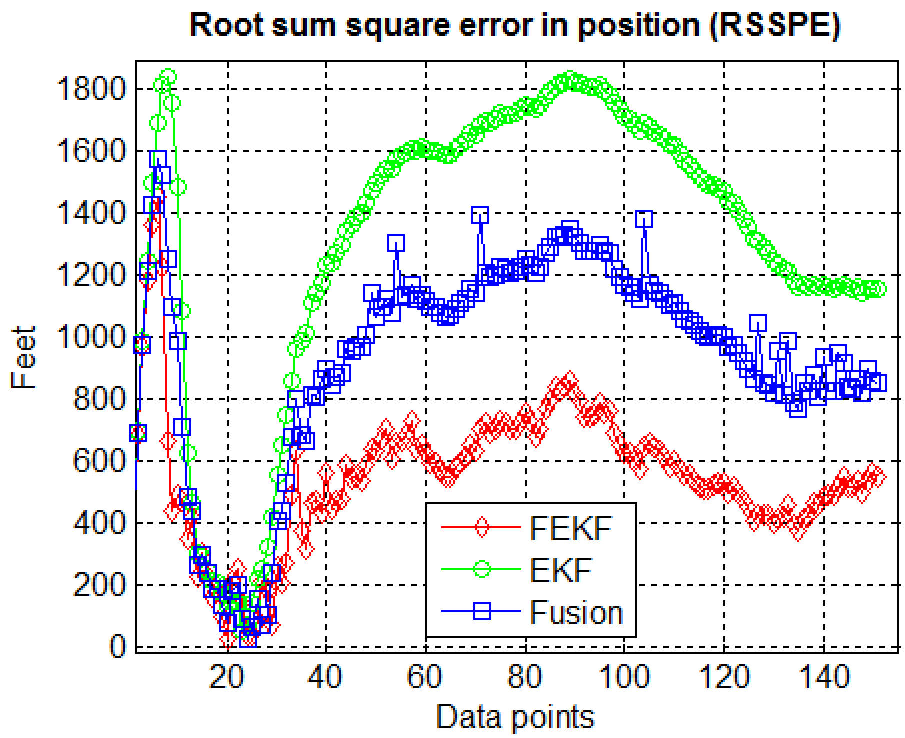

Figure 4. Root sum square error in position of the estimated trajectories by Kalman filters and Data fusion

Unlike the Example 1 in this example we have less confidence in the initial position and velocity in the Y coordinate, and as we can see, the convergence to the actual trajectory is much faster, obtaining a RSSPE lower for all the filters.

The following table summarizes the highlights from the previous example; it shows the Percentage fit error (PFE) in position of each filter

Table 2: PFE in Position

Filters	X-position	Y-position
FEKF	4.4804	7.0362
EKF	11.5410	15.6906
Data Fusion	8.0950	11.3819

It can be seen in the Table 2 that the FEKF again gives a more accurate estimate than the EKF and the data fusion, But the FEKF does not always give better performance than the EKF in scenarios with different initial values, due to the rules *if…Then* in the FIS system could not consider all event information.

4. Conclutions

The application of Fuzzy Logic to the Kalman Filtering and Data Fusion give a more robust implementation of target tracking application, because of the use of rules for the inference in the fuzzy inference system, which are generally created based on the experience and intuition of the domain expert. Moreover the advantage of using data fusion is that it reduces uncertainty in the perception of scene by a single sensor, because of multiple sensors would provide redundancy which, in turn, would enable the system to provide information in case of partial failure or data loss from one sensor.

REFERENCES

Blackman, S. 1986. Multiple target tracking with radar applications . Dedham, MA: Artech House.

Dan Simon, Optimal State Estimation Kalman, H-infinite, and Nonlinear Approaches, John Wiley & Sons, pp-456.

Egfin Nirmala, D., V. Vaidehi, S. Indira Gandhi. 2005. Data fusion using fuzzy logic for multi target tracking. In Proceedings of International Radar Symposium India, pp. 75–80.

Jitendra R. Raol, Multi-Sensor Data Fusion with MATLAB, CRC Press, pp-332-333, 338-341.

Kashyap, S. K., and J. R. Raol. 2008. Fuzzy logic applications for filtering and fusion for target tracking. Def Sci J 58:120–35.

Klein, L. A. 2004. Sensor and data fusion: A tool for information assessment and decision making. Washington, DC: SPIE Press.

Raol, J. R., and J. Singh. 2009. Flight mechanics modeling and analysis . Boca Raton, FL: CRC Press.

Wilfried, E. 2002. An introduction to sensor fusion. Research report 47/2001 . Austria: Institute fur Technische Informatik, Vienna University of Technology.

Authorization and Disclaimer