Fuzzy Logic Applications in Filtering and Fusion for Target Tracking

S.K. Kashyap and J.R. Raol

National Aerospace Laboratories, Bangalore–560 017

ABSTRACT

A fuzzy Kalman filter algorithm is developed for target tracking applications and its performance evaluated using several numerical examples. The approach is relatively novel. A comparison with Kalman filter and an adaptive tuning algorithm is carried out. The applicability and usefulness of fuzzy logic in data fusion is also demonstrated. The performance of both the extended Kalman filter and fuzzy extended Kalman filter is evaluated using real data of a manoeuvering target and it is found that fuzzy extended Kalman filter shows better performance as compared to extended Kalman filter.

Keywords: Target tracking, extended Kalman filter, adaptive tuning algorithm, fuzzy logic, data fusion, fuzzy Kalman filter, performance evaluation, state vector fusion

1. INTRODUCTION

It is essential to get accurate information about target states such as position, velocity, and acceleration from the noisy measurements originating from single source or multiple sources. Kalman filter (KF) is a suitable algorithm for such applications. In case of multiple sources, either single KF can be used by fusing the measurements at data level or by state vector fusion (SVF). In case of nonlinear system and measurement models, extended KF (EKF) is used wherein dynamics are linearised wrt predicted/estimated system states. The accuracy of estimated/fused states depends upon: (i) how accurate the target and measurement models are, and (ii) process noise covariance Q and measurement noise covariance R that basically decide the bandwidth of a filter. In many situations, mathematical models are not known accurately or difficult to obtain. In practice, modelling errors are compensated by tuning the filter, for Q, using trial and error or some heuristic approach.

A proper combination of fuzzy logic (FL) and KF and a fuzzy Kalman filter (FKF) is investigated for target tracking applications. The performances of KF and FKF are compared with adaptive Kalman filter (AKF) in which process noise covariance Q is computed online using sliding window method. Also, the applicability and usefulness of FL in data fusion is demonstrated and its performance compared with conventional method of fusion, i.e., SVF.

2. FUZZY LOGIC-BASED KALMAN FILTER

The KF is suitable for real-time target tracking. The filter has some inherent soft decision-making ability that decides on how much emphasis should be given to target models or the measurement data. The decision-making is controlled through the filter gain (Kalman gain) that depends upon the ratio of the strength of Q and R. The KF algorithm in discrete-time domain is as follows:

(a) *System models*

$$X(k+1) = FX(k) + Gw(k) \qquad (1)$$

$$Z_m(k) = HX(k) + v(k) \qquad (2)$$

(b) *State and covariance time propagation*

$$\hat{X}(k+1/k) = F\hat{X}(k/k) \qquad (3)$$

$$\hat{P}(k+1/k) = F\,\hat{P}(k/k)\,F^T + GQG^T \qquad (4)$$

(c) *State and covariance (measurement) update*

$$\left. \begin{aligned} K &= \hat{P}(k+1/k)H^T S^{-1} \\ S &= H\hat{P}(k+1/k)H^T + R \end{aligned} \right\} \qquad (5)$$

$$\left. \begin{aligned} \hat{X}(k+1/k+1) &= \hat{X}(k+1/k) + Ke(k+1) \\ e(k+1) &= Z_m(k+1) - H\,\hat{X}(k+1/k) \end{aligned} \right\} \qquad (6)$$

$$\hat{P}(k+1/k+1) = [I-KH]\hat{P}(k+1/k) \qquad (7)$$

where $X(k)$ is the n-dimensional state, F is the state transition matrix, G is the process noise gain matrix, $w(k)$ is the process noise (white and Gaussian) with zero mean with covariance matrix Q, and k is the scan number. $Z_m(k)$ is the m-dimensional measurement vector, H is, m x n, observation matrix, $v(k)$ is the measurement noise with Gaussian distribution and zero mean with a covariance matrix R. $\hat{X}$ is predicted state vector, $\hat{P}$ is predicted state covariance matrix, K is Kalman gain, S is innovation covariance matrix, e is innovation sequence vector, $\hat{P}$ is estimated state vector, and $\hat{P}$ is estimated state covariance matrix.

Fuzzy logic is a multi-value logic[1-3] used to model any event or condition that is not precisely defined or known. In the FL-based system, one uses: (i) membership function that converts the input/output crisp values to corresponding membership grades indicating its belongingness to respective fuzzy set, (ii) rule base consisting of IF-THEN rules, (iii) fuzzy implications maps the fuzzified input to appropriate fuzzified outputs, (iv) aggregation to combine the output fuzzy sets (single output fuzzy set for every rule fired) to single fuzzy set, and (v) defuzzification to convert aggregated output fuzzy set from its fuzzified values to equivalent crisp values. In a KF, since the innovation sequence is the difference between the sensor measurement and the predicted value based on filter's model, this mismatch can be used to perform the required adaptation using fuzzy logic rules[4]. The advantages derived using the fuzzy technique are the simplicity of the approach, the possibility of accommodating the heuristic knowledge about the phenomenon, and the relaxation of some of the *a priori* assumptions of the process. This aspect is accommodated in Eqn (6) as given by

$$\hat{X}(k+1/k+1) = \hat{X}(k+1/k) + KC(k+1) \qquad (8)$$

where $C(k+1)$ is the fuzzy correlation variable[5] (FCV) and is a nonlinear function of the innovations. To find $C(k+1)$, the innovation vector e is first separated into its x and y components, for 2-D target tracking applications: e_x and e_y. It is assumed that target motion in each axis is independent. The FCV consists of two inputs (i.e., e_x and $\dot{e}_x$) and single output $c_x(k + 1)$, where $\dot{e}_x$ is computed by

$$\dot{e}_x = \frac{e_x(k+1) - e_x(k)}{T} \qquad (9)$$

where, T is the sampling time interval in seconds. Interestingly, the FKF can be used to track a manoeuvering or a non-manoeuvering target. In any fuzzy inference system (FIS), (Fig. 1), fuzzy implication provides mapping between input and output fuzzy sets. Basically, a fuzzy IF-THEN rule
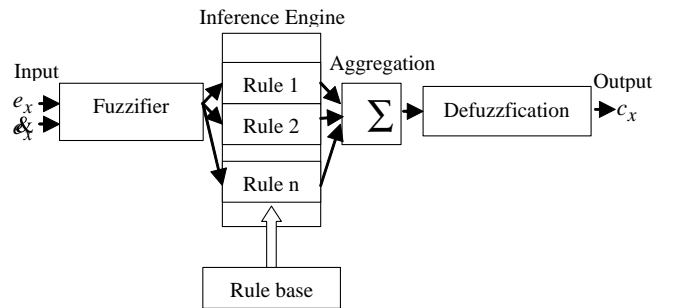

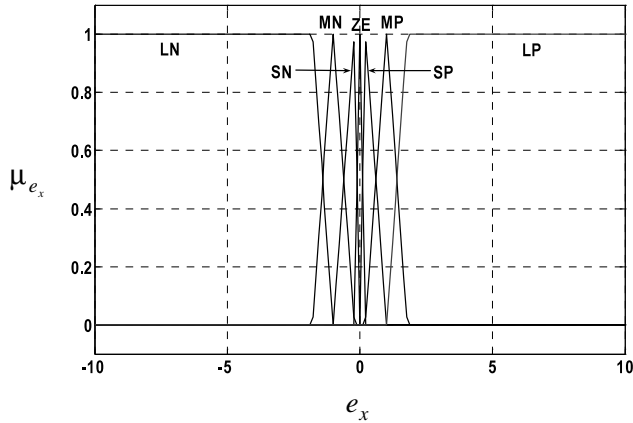**Figure 1. Schematic of a fuzzy inference system.**

Figure 2. Membership functions for input e_x (μ indicates degree of membership).
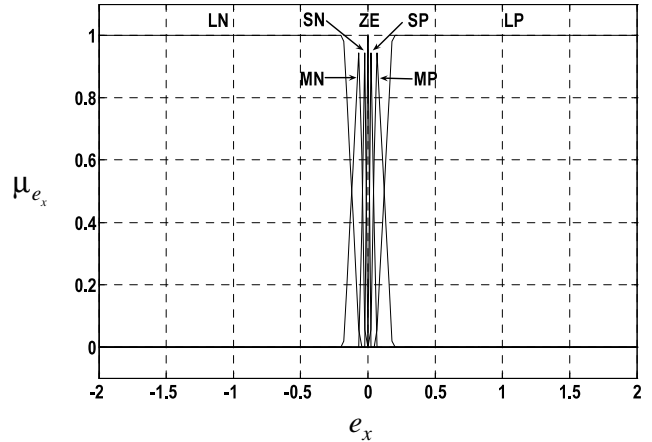

Figure 3. Membership functions for input $\dot{e}_x$ (m indicates degree of membership).

is interpreted as a fuzzy implication. The antecedent membership functions that define the fuzzy values for input e_x and $\dot{e}_x$, are shown in Figs 2 and 3, respectively. Similarly membership functions for output c_x are shown in Fig. 4. The labels used in linguistic variables to define membership functions are LN (large negative), MN (medium negative), SN (small negative), ZE (zero error), SP (small positive), MP (medium positive), and LP (large positive). The rules for the inference in FIS are created based on the past experiences and intuitions. For example, one such rule is:

$$\text{IF } (e_x \text{ is LP}) \text{ AND } (\dot{e}_x \text{ is LP}) \text{ THEN } c_x \text{ is LP} \quad (10)$$

This rule is created based on the fact that having e_x and $\dot{e}_x$ with large positive values indicates an increase in innovation sequence at a faster rate. The future value of e_x (and therefore $\dot{e}_x$) can be reduced by increasing the present value of c_x (some function of $\approx Z \quad H\chi^{\%}$) with a large magnitude. Table 1 gives 49 rules[5] needed to implement FCV. Output c_x at any instant of time can be computed

using the input e_x and $\dot{e}_x$, input membership functions, rules mentioned in Table 1, FIE, aggregator, and defuzzification. Figure 5 illustrates the 3-D surface view of FIS for present application. The properties of FIS used in the present work are: (i) FIS type: mamdani, (ii) AND operator: min, (iii) OR operator: max, (iv) fuzzy implication method: min, (v) aggregation method: max, and (vi) defuzzification method: centroid.
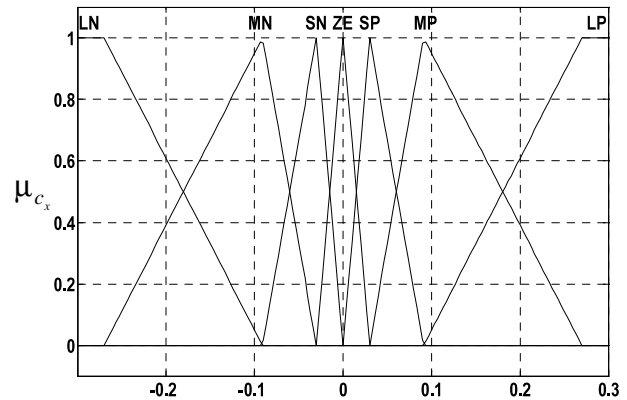

Figure 4. Membership functions for output c_x.

Table 1. Fuzzy associated memory for output C_x with 49 rules

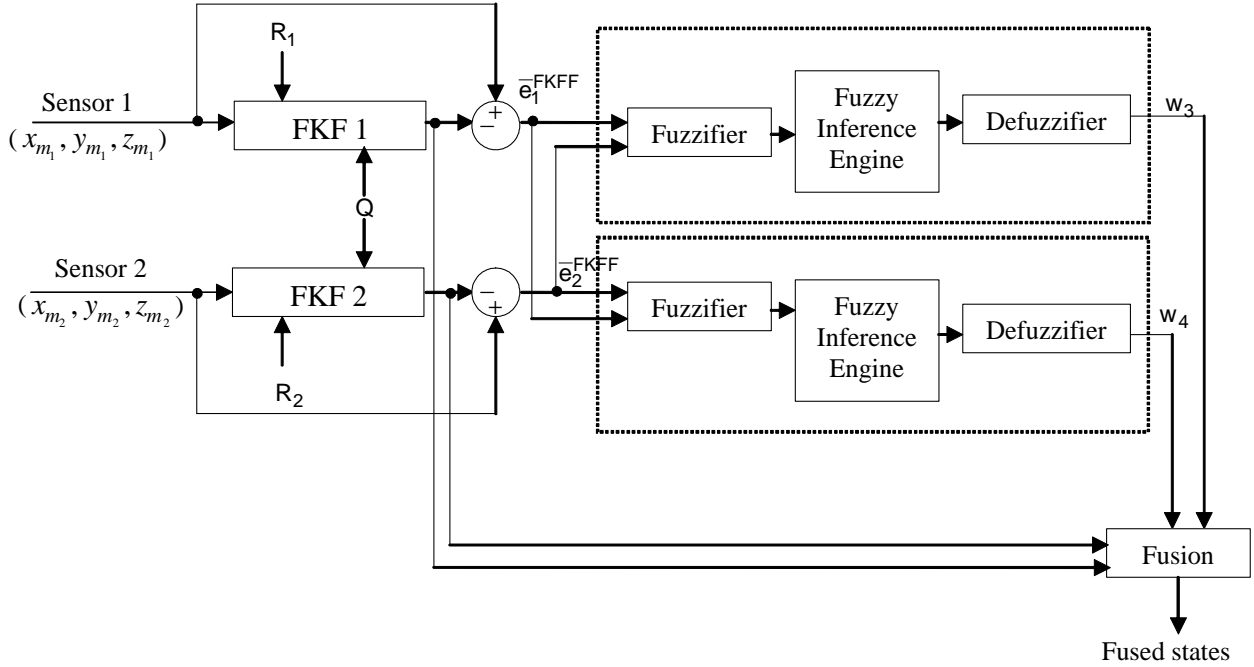| $\dot{e}_x$	e_x						
	LN	**MN**	**SN**	**ZE**	**SP**	**MP**	**LP**
LN	LN	LN	MN	MN	MN	SN	ZE
MN	LN	MN	MN	MN	SN	ZE	SP
SN	MN	MN	MN	SN	ZE	SP	MP
ZE	MN	MN	SN	ZE	SP	MP	MP
SP	MN	SN	ZE	SP	MP	MP	MP
MP	SN	ZE	SP	MP	MP	MP	LP
LP	ZE	SP	MP	MP	MP	LP	LP

Figure 8. Proposed Scheme for FKFF method.

- In Eqns (13) and (14), superscript KF is replaced by FKF, which means that state estimation is performed using FKF instead of KF.

- The fused states are obtained by modifying Eqn (15), given as under

$$\hat{X}_f^{FKFF}(k) = \hat{X}_1^{FKF}(k) \quad w_3(k)\left(w_3(k) \quad w_4(k)\right)^{-1}$$
$$\left(\hat{X}_2^{FKF}(k) - \hat{X}_1^{FKF}(k)\right) \quad (16)$$

The point to be noted here is that Eqn (16) is obtained from Eqn (11) by replacing $\hat{P}_1^{KF}$, $\hat{P}_2^{KF}$ with w_3 and w_4, respectively.

4. VALIDATION OF VARIOUS FILTERING/ FUSION SCHEMES

4.1 Target Tracking

The target data in x-direction is generated using constant acceleration model with process noise increment. With sampling interval $T = 0.1$ s, a total of $N = 100$ scans are generated. The data simulation proceeds using target state [Eqn (1)] and measurement [Eqn (2)] with the following assumed parameter values: (a) initial states of target ($x, \dot{x}, \ddot{x}$) are (0 m,

100 m/s, 0 m/s²) respectively, (b) process noise variance $Q = 0.0001$, and (c) measurement noise variance $R = 10$, and measurement vector $H = [1\ 0\ 0]$, and constant acceleration model given below:

$$F = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \quad (17)$$

$$G = \begin{bmatrix} T^3/6 & T^2/2 & T \end{bmatrix} \quad (18)$$

The initial conditions, F, G, H, Q, and R for both the filters are kept the same. The initial state vector estimate $\hat{X}(0/0)$ is kept close to initial state vector of simulated model. The results for both the filters are compared in terms of true versus estimated states and states errors with bounds at every scan number. Figure 9 shows the comparison of true and estimated positions, velocities and accelerations of target in x-direction. Every effort was made to tune the KF properly. It is clear from the plots that KF shows initial transient and takes more time to settle down as compared to FKF. Figure 10 illustrates the comparison of state errors which are within $\pm 2\sqrt{\hat{P}(i,i)}$ bounds for both the filters. Evidently, it is observed that FKF performs much better compared to KF.
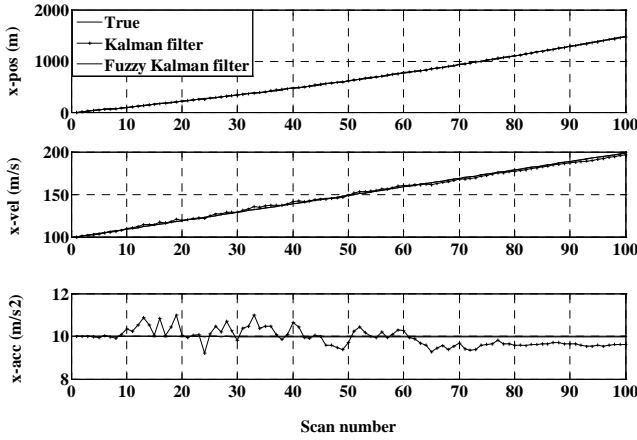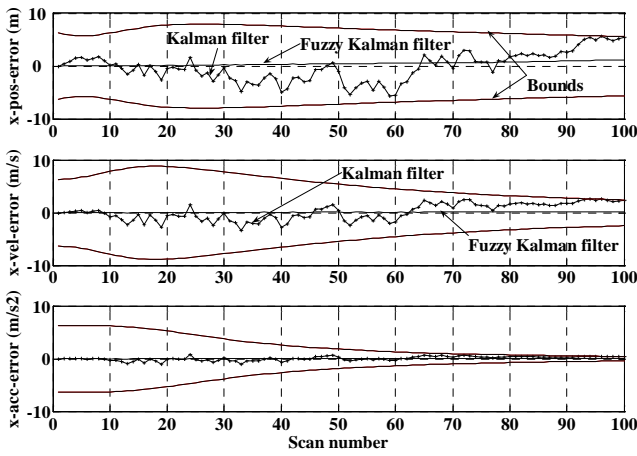
Figure 9. Comparison of true and estimated states.

Figure 10. States errors with bounds.

4.1.1 Consistency Check of KF and FKF

The normalised cost function (CF) is computed using the following formula:

$$CF = \frac{1}{N}\sum_{k=1}^{N} e(k)S(k)^{-1}e(k)^T \qquad (19)$$

Here, e is innovation sequence vector. The filter performance is consistent when its normalised CF in Eqn (19) is equal to the dimension of measurement vector. The consistency check of KF and FKF is done for the following: (a) case 1: only position as measurement ($R = 100$), (b) case 2: only position and velocity as measurements [$R = \text{diag}(100,1)$], and (c) case 3: only position, velocity and acceleration as measurements [$R = \text{diag}(100,1,0.01)$]. It is observed, from Table 4, that the cost function for KF is close

Table 4. Comparison of cost function for KF and FKF

	Normalised cost function		
	True	Computed	
		KF	FKF
Case 1	1	0.93	0.85
Case 2	2	1.96	1.84
Case 3	3	2.94	2.85

to the theoretical value, therefore filter performance is consistent. For FKF, its cost function is slightly away from the theoretical prediction but still comparable with KF. This means that FKF is approximately a consistent filter.

Figures 11-13 illustrate position, velocity, and acceleration errors. From these results one observed: (a) for KF the state error reduces when additional observables (like velocity) are used as measurements, (b) for FKF even the use of only position measurements give considerable reduction in state error compared to KF, (c) FKF shows overall better performance than KF for all the three cases, and (d) since, for KF and FKF, the theoretical innovations covariance happens to be the same, the CF for FKF is lower than that of KF (since the computed innovations of FKF gives lower variance than the KF). This requires further study.

Both the filter schemes are also evaluated for the state estimation of target data in x- and y-directions. It is assumed that target motion in each axis is independent. With that assumption, target data in y- direction is simulated (using the same models/parameters as used in x-direction) with an initial states of $y, \dot{y}, \ddot{y}$ = [0 m, –100 m/s, –10 m/s²]. Due to inclusion of y-direction data, matrices such as F, G, H, and R used is both the filter are as follows:

$$F = \begin{bmatrix} 1 & T & T^2/2 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & T^2/2 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (20)$$
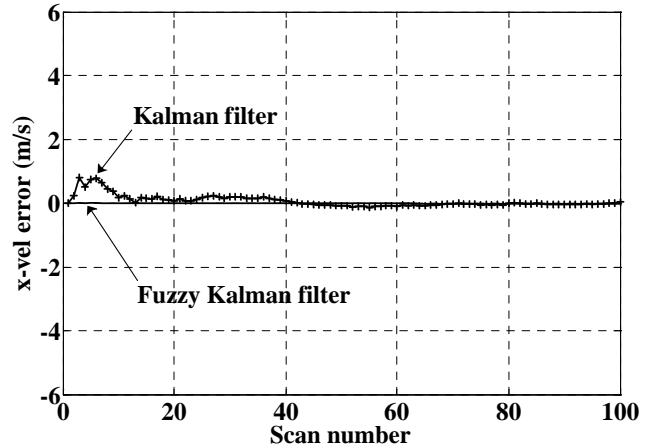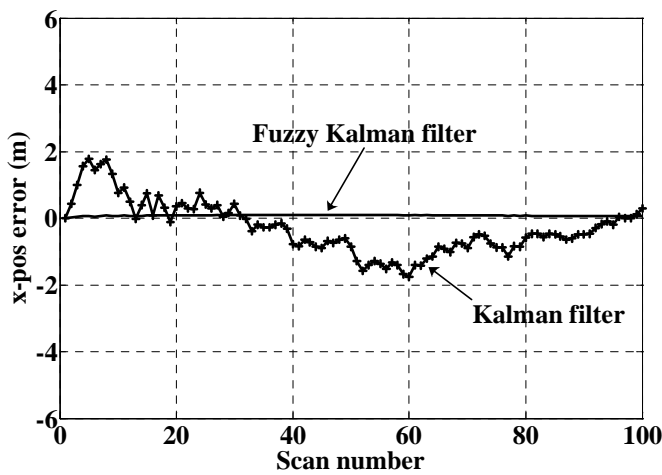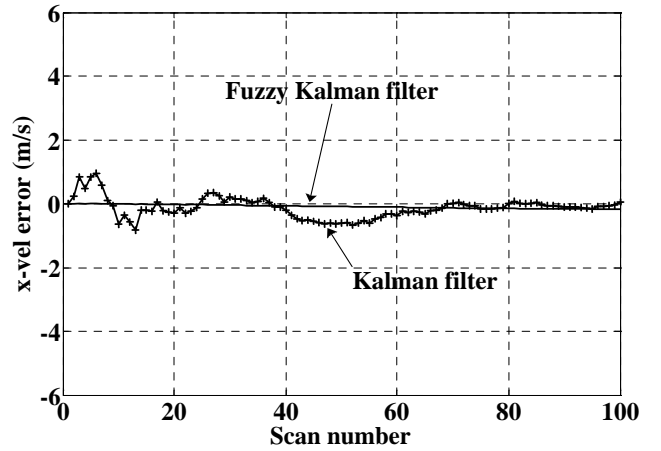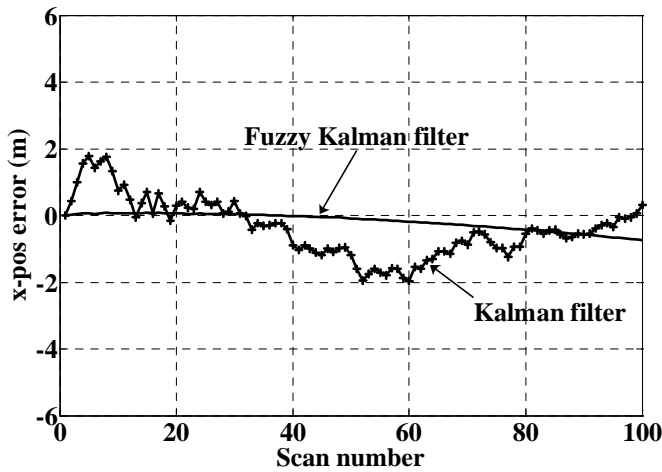
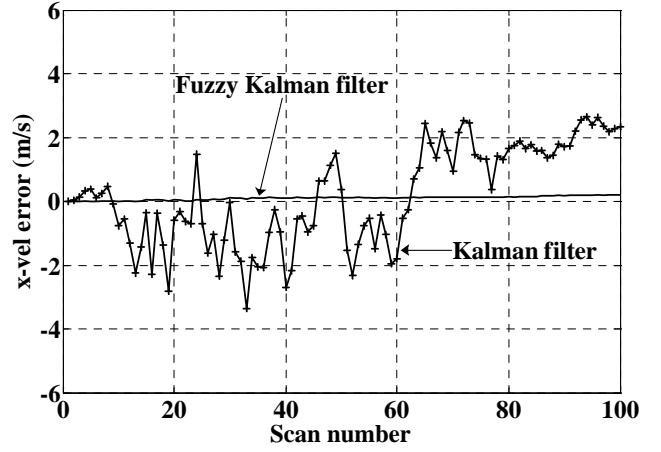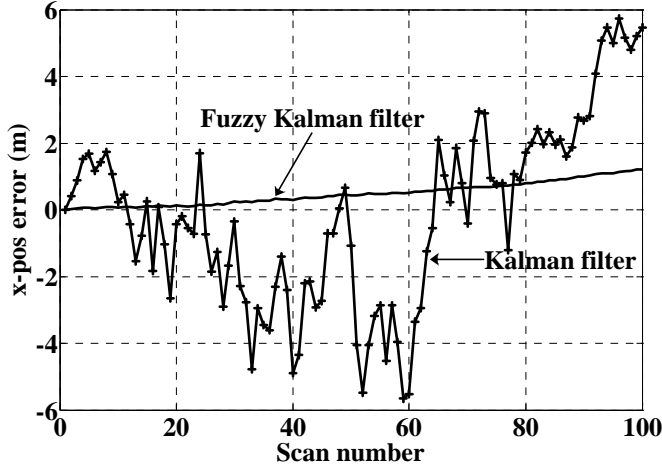Figure 11. Comparison of position error for both the filters.

Figure 12. Comparison of velocity error for both the filters.

$$G = \begin{bmatrix} T^3/6 & T^2/2 & T & T^3/6 & T^2/2 & T \end{bmatrix} \quad (21)$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (22)$$

$$R = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix} \quad (23)$$

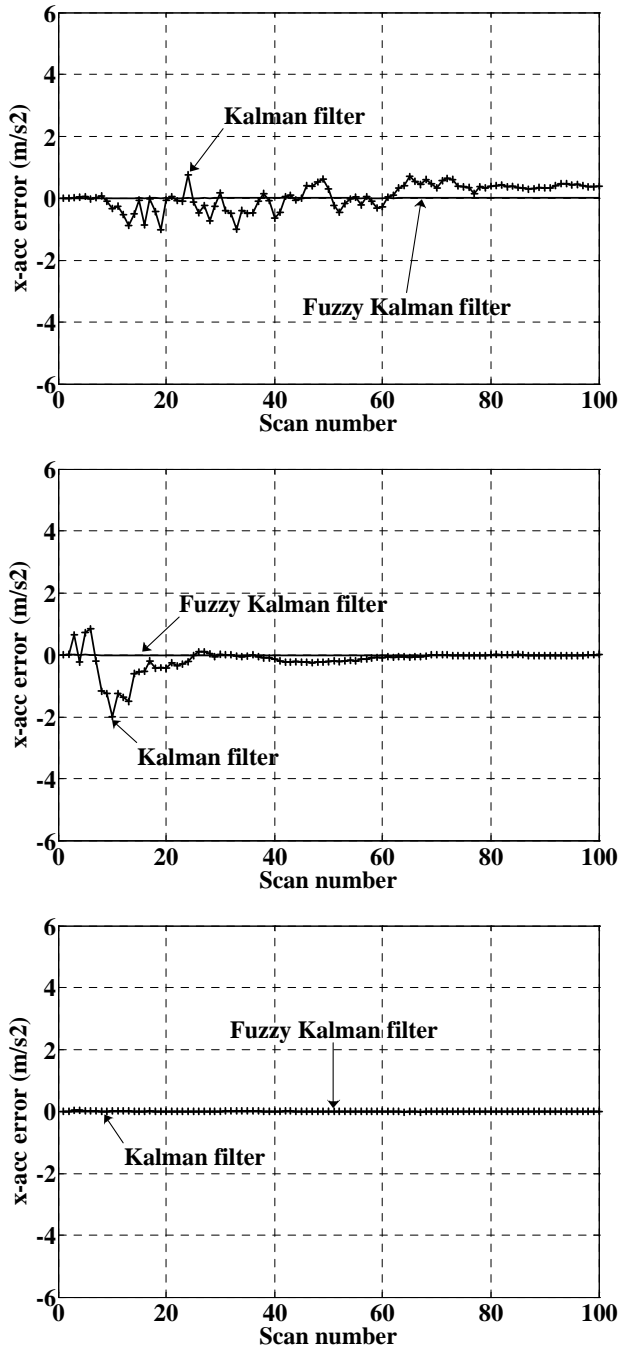The same FCV is used for *y*-direction. For *y*-axis, similar performance (as reported in Figs 9-10)

127

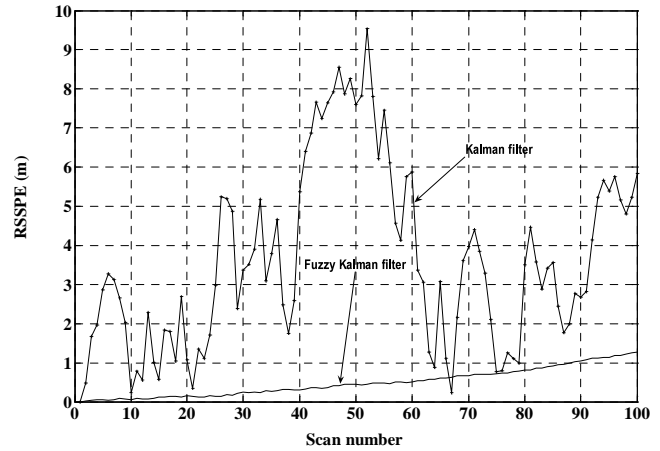**Figure 13. Comparison of acceleration error for both the filters**

Figure 14. Comparison of RSSPE for both the filters.

Figures 14-16 compare the RSSPE, RSSVE, and RSSAE computed using true and estimated states for both the filters. Although the performance of the KF is satisfactory and acceptable, the FKF performs better than the KF. For the same target data (i.e. x and y axes), the performance of KF with FKF is compared for two cases: (i) when all the 49 rules are taken into consideration (Table 1), and (ii) only 4 rules are used (Table 5). The membership functions for new FCV are shown in Figs 17-19. Figure 20 shows the 3-D surface view of input-output mapping for new FCV. Figure 21 compares the RSSPE of FKF for these two cases. The filter with 49 rules shows better performance than the filter with only 4 rules, although the performance with 4 rules is acceptable. Thus to have a good FKF, just a sufficient number of rules is needed to get continuous and smooth input-output mapping.

4.2 Tracking of Manoeuvering Target

For tracking manoeuvering target, it is essential to re-design the FCV to capture the various possible manoeuver modes of the target. Re-designing of FCV involves: (a) proper selection of membership functions of inputs and output, (b) tuning of selected

is observed for both the filters, therefore, it is not shown again. The performances of both the schemes are also compared in terms of RSSPE [root sum square position error

$$\text{RSSPE} = \sqrt{(x(k/k)-\hat{x}(k/k))^2 \quad (y(k/k) \quad \hat{y}(k/k))^2}\ ].$$

Table 5. Fuzzy associated memory with 4 rules

$\&_x$	e_x	
	LN	**LP**
LN	LN	ZE
LP	ZE	LP

Case A: *Mild Manoeuver Data*

To generate mild manoeuver data, the previous data were used with minor modification in arbitrary acceleration injection points. A total of 17 scans were generated. Accelerations were injected at scans 8 (x_{acc}= 6 m/s^2 and y_{acc}= –6 m/s^2) and 15 (x_{acc}= –6 m/s^2 and y_{acc}= 6 m/s^2) only.

Case B: *Evasive Manoeuver Data*

To generate evasive manoeuver data, the same points for arbitrary acceleration injection but with a manoeuver magnitude of 40*9.8 m/s^2 (i.e., instead of 9*9.8 m/s^2) are used. The results for these cases are obtained for 100 Monte-Carlo runs. The initial state vectors of KF and FKF are the same and kept close to initial true simulated states. Initial state error covariance matrices for both the filters are kept to unity.

Figure 23 compares the measured, true, and estimated *x*-y target positions for *Case A*. The comparison of estimated trajectories from KF and FKF with true and measured trajectory is reasonably good. Some discrepancies are observed in the manoeuvering phase of the flight where FKF exhibits better performance than KF. Similar observations are made for *Case B* (results not shown). In Fig. 24 of *Case B*, the RSSPE, RSSVE, and RSSAE for KF are found to be large compared to those for FKF, indicating that KF is unable to satisfactorily

track the target during the manoeuver phase compared to FKF.

4.3 Comparison of FKF, KF, and Adaptive KF

The performance of FKF is compared with KF and adaptive KF (AKF) using simulated data of *Case A*: mild manoeuver. The equations of AKF are the same as those of KF (mentioned in Section 2) but with varying process noise covariance *Q*, estimated online using Maybeck method[8]. The equations required to estimated *Q* are given by

$$Q(k) = G^{\#}\left[\hat{P}^{-+} - F\hat{P}\ F^T\right](G^{\#})^T \tag{30}$$

$$\hat{P}^- = K(k)\hat{A}(k)*(H^T)^{\#} \tag{31}$$

$$\hat{P}^+ = \hat{P} - K(k)H\hat{P} \tag{32}$$

$$\hat{A}(k) = \frac{1}{WL}\sum_{j=k+WL-1}^{k} e(j)e(j)^T \ ; \ k \geq WL \tag{33}$$

Here, # stands for pseudo-inverse, *e* is the innovation sequence vector computed using Eqn.(4), *k* is the scan number and *WL* (=5 for the present case) is the window length. It is important to note that online value of *Q* can be made available to AKF only from *WL*[th] scan which means that accuracy of *Q* will depend upon its initial guess(for
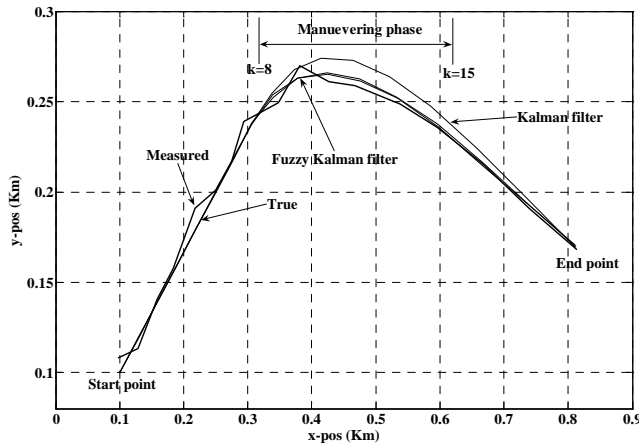

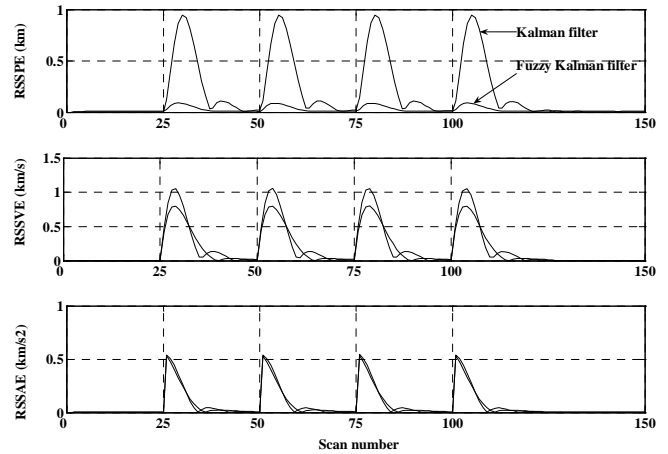Figure 23. Comparison of true, measured and estimated x-y target positions-mild manoeuver.

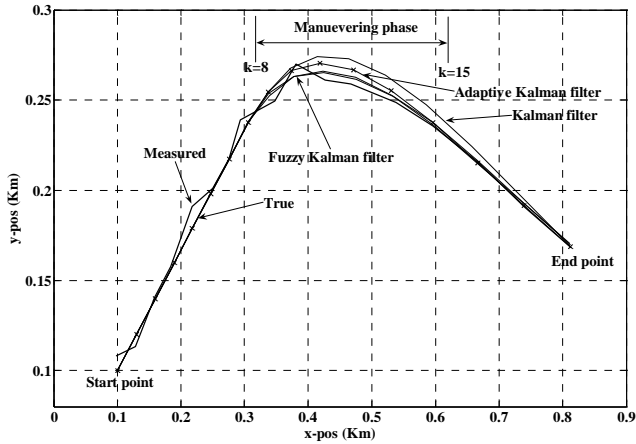Figure 24. Comparison of RSSPE, RSSVE, and RSSAE for both the filters-evasive manoeuver.

Figure 25. Comparison of true, measured, and estimated *x-y* target positions.

Figure 26. Comparison of RSSPE.

$k = 1$ to $WL^{th} - 1$ scans) chosen by a filter designer. Figure 25 compares the measured, simulated true, and estimated *x-y* target positions. It is clear that AKF exhibits better tracking accuracy, especially during target manoeuver, as compared to KF but still it has slightly degraded performance as compared to FKF. For non-manoeuvering phases of target motion, AKF and KF perform almost similar. During the manoeuvering portion, it is found that for AKF, the magnitude of online Q increases, and hence, also Kalman gain, which in turn assigns more weight to measurement model and thus aids in convergence of the estimated states to true values at a much faster rate than as seen for KF only.

Additionally, a sensitivity study of AKF wrt different values of *WL* (5 and 10) was carried out and its performance in terms of RSSPE (Fig. 26) was compared with KF and FKF. It can be observed that RSSPE of AKF during manoeuvering phase is high for *WL* = 10 compared to for *WL* = 5. This could be due to non-availability of online Q and with an assumption that its initial guess is not appropriate at a time when actual manoeuver starts i.e., at $k = 8$. An appropriate selection of *WL* requires *a priori* knowledge about when the target will manoeuver first (that may not be available for an enemy target) puts a limitation on AKF as compared to FKF.
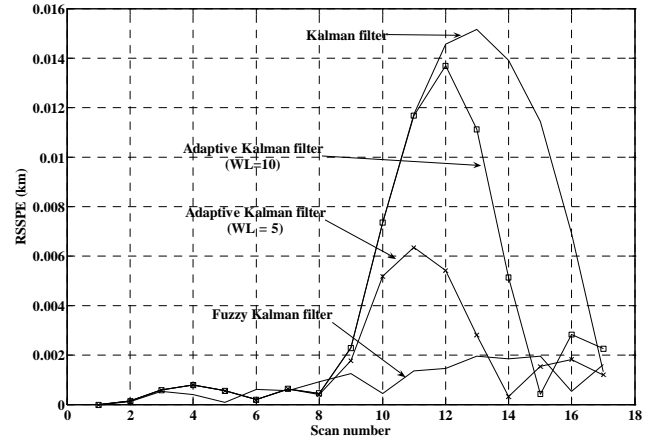
4.4 Fusion

To compare the performance of various fusion algorithms such as SVF, KFF, and FKFF, new data are generated by modifying the arbitrary acceleration injection points used to generate previous data. For a new data set, a total of 25 scans were generated. Accelerations were injected at scans 8 (x_{acc} = 6 m/s^2 and y_{acc} = –6 m/s^2) and 15 (x_{acc} = –6 m/s^2 and y_{acc} = 6 m/s^2) only. The measurements for two sensors were generated with SNR of 10 for sensor 1 and SNR of 20 for sensor 2. The measurements from each sensor were processed by KF and FKF (tuned FCV obtained in Section 4.2 is used here) for 100 Monte-Carlo runs. The initial states (80 % of true initial state) and error covariance were kept the same. Figure 27 shows the comparison of true *x-y* target positions with estimated positions obtained using SVF, KFF, and FKFF methods. The comparison of estimated trajectories from three methods with simulated true trajectory is reasonably good, especially before the target first manoeuvres (i.e., $0 < k < 8$). Some discrepancies were observed in the manoeuvering phase of the flight where FKFF exhibited better performance than SVF and KFF. Figures 28-29 illustrate the position error comparisons for: (i) SVF, FKFF, and (ii) KFF, FKFF. Figures 30-31 compare the velocity and acceleration error, respectively.
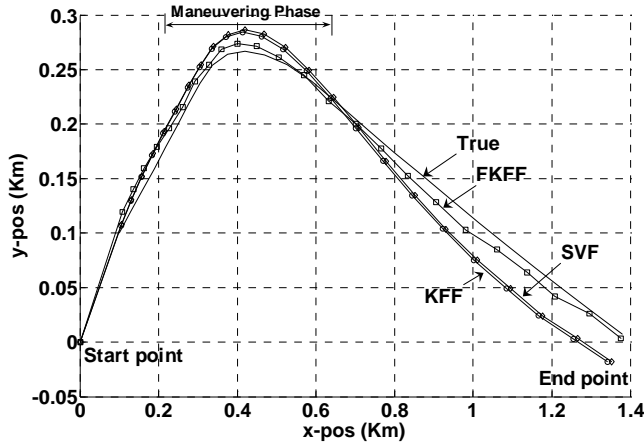
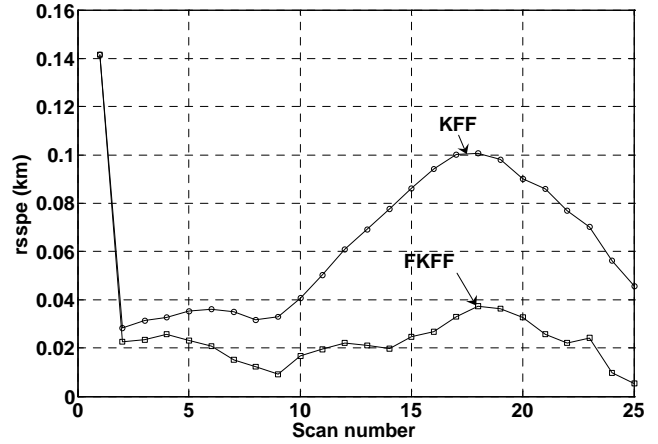Figure 27. True and estimated fused trajectories.

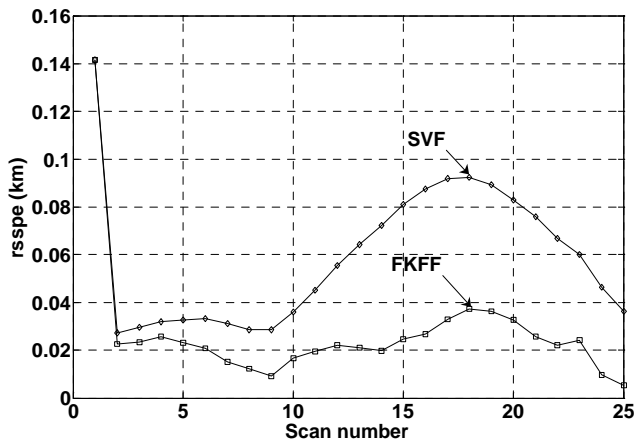
Figure 29. RSSPE - KFF/FKFF.
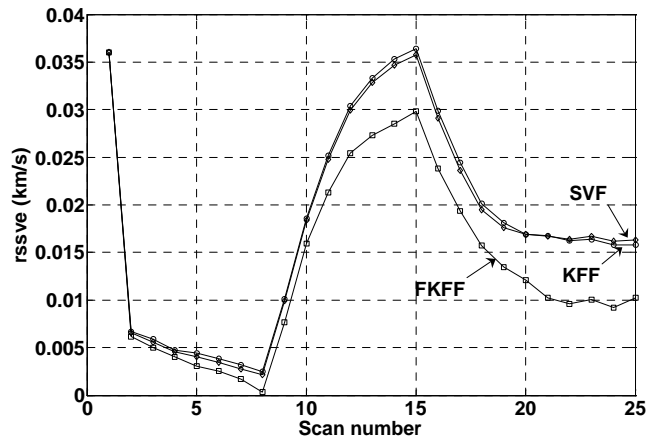

Figure 28. RSSPE - SVF/FKFF.

Figure 30. RSSVE - SVF/KFF/FKFF.

Following observations are made: (i) FKFF results are better than those of SVF and KFF, (ii) during manoeuvering, portion FKFF shows overall less state error as compared to rest of fusion schemes, and (iii) since FKFF method depends directly on the empirical errors, it is expected to give better accuracy as compared to SVF method in which state error covariance matrices are based on prediction given out by the KF.

Figure 32 shows the computed weights for KFF and FKFF methods. It can be observed from the top two subplots that initial values of w_1 and w_2 are comparable but as more number of data points are processed, value of w_2 becomes larger than w_1, which simply means that lesser weight is assigned to estimated states from sensor 1 as compared to sensor 2. This is logically correct because sensor 1 gives nosier measurements (SNR=10) as compared

to sensor 2 (SNR=20). The bottom sub-plot of Fig. 32 shows the summed weights and it is observed that values are around 1 (as expected) for KFF and FKFF methods.

4.5 Real Data Analysis using Fuzzy EKF

The performance of EKF[1] and FEKF was compared using real data. The real data consists of target information in polar frame measured by a sensor. The state estimation, consisting of target's position, velocity, and acceleration information, was carried out in 3-D Cartesian frame using constant acceleration model and polar data. The sampling time interval chosen was 0.25 s with process noise covariance Q kept at 0.0001*eye(9,9), whereas, measurement noise covariance value R was obtained from sensor specification. The initial state of EKF and FEKF was kept close to simulated true initial
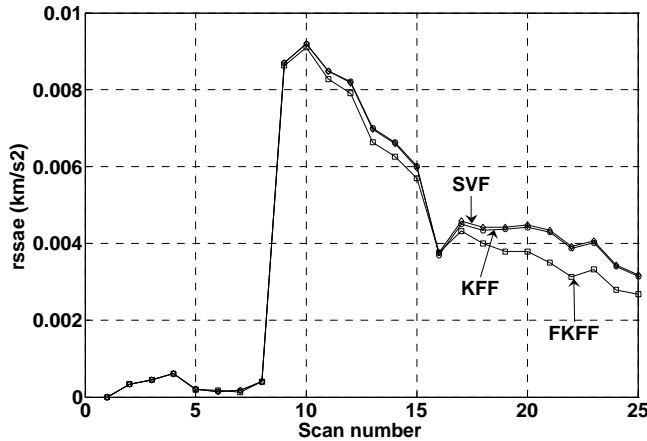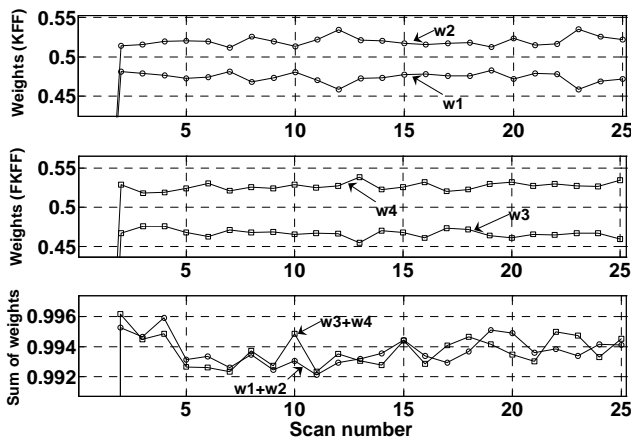
Figure 31. RSSAE - SVF/KFF/FKFF.

Figure 32. Weights and its sum - KFF/FKFF.

target state and accordingly initial state error covariance is computed. The FCV for FEKF is kept the same as used in Section 4.2. It is clear form the Fig. 33 that by in large RSSPE of EKF is comparable with that of FEKF except at few points where target exhibits manoeuvers. At these points RSSPE was higher for EKF. Similar observations were made for RSSVE and RSSAE errors.

5.   CONCLUSIONS

Several variants of fuzzy logic-based Kalman filters and fusion schemes are evaluated using simulation data for target tracking. To track manoeuvering target, FCV is re-designed using training and checking data sets obtained from simulated true and measured target positions. It is clear that FKF gives comparatively better performance than KF. It is suggested that just sufficient number of
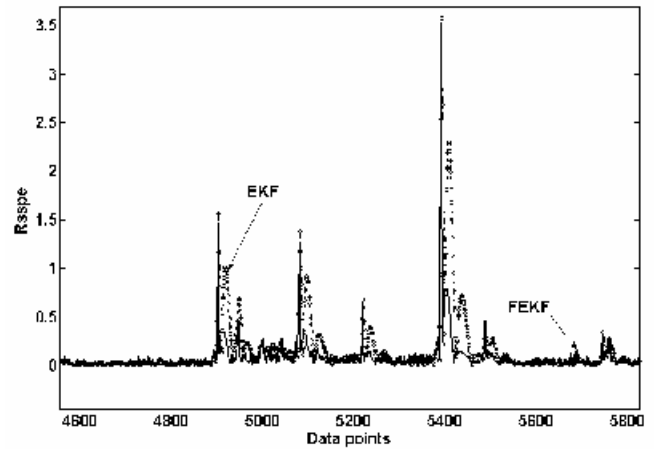

Figure 33. RSSPE comparison of EKF and FEKF.

rules could be used rather than a large number of rules to develop an efficient FCV, and in turn, good FKF. KF, and FKF are compared with AKF and it is found that although AKF shows better result as compared to KF but still not as good as FKF, especially during manoeuvering phase of the target. The fused trajectories obtained using SVF, KFF, and FKFF are compared and it is found that FKFF shows overall better performance as compared to SVF and KFF algorithms. The performance of EKF and FEKF is compared using real data and it is found that FEKF yields better estimates than EKF, especially in the region where target manoeuvers.

REFERENCES

1. Passino, K.M. & Yurkovich S. Fuzzy control. Addison-Wesley, 1998.

2. Kashyap, S.K. & Raol, J.R. Unification and interpretation of fuzzy set operations. *In* CCECE-CCGEI, IEEE Canadian Conference on Electrical and Computer Engineering, Ottawa, Canada, May 2006. pp. 353-58.

3. Kashyap, S.K. & Raol, J.R. Interpretation and unification of fuzzy set operations and implications, accepted for publication in *Journal of Systems, Science and Engineering* (JSSE), Systems Society of India, VSSC, Thiruvananthapuram, India.

4. Raol, J.R.; Girija, G. & Singh, Jatinder. Modelling and parameter estimation of dynamic systems.

In IEE Control Engineering Series Book, Vol. 65 IEE, London, August 2004.

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

6. David, L. Hall & Llinas, James. An introduction to multisensor data fusion. *Proceedings IEEE*, January 1997, **85**(1).

7. Nirmala, D. Egfin; Vaidehi, V. & Gandhi, S. Indira. Data fusion using fuzzy logic for multi target tracking. *In* Proceeding of International Radar Symposium India, 19-22 December 2005. pp. 75-80

8. Peter, S. Maybeck. Stochastic models, estimation, and control, Vol. 2. Academic Press Inc., London Ltd., 1982.

Contributors

Mr Sudesh K. Kashyap obtained ME (Electrical Engg) with specialisation in automatic control and robotics from MS University of Baroda in 1999. He is working at National Aerospace Laboratories (NAL), Bangalore, as Scientist in the area of multi-sensor data fusion. He is a recipient of *Young Scientist Award -2007* at NAL, Bangalore. His areas of interest are control systems, estimation, data fusion, fuzzy logic, and neural networks.
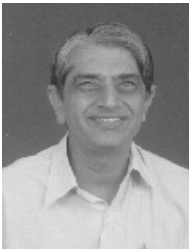
Dr J.R. Raol obtained his ME from the MS University, Baroda in 1974 and PhD from the McMaster University, Canada, in 1986. He worked at NAL, Bangalore, and was actively involved in the multidisciplinary control group's activities on human pilot modelling in fixed- and motion-based research simulators. He served as the Head of the Flight Mechanics and Control Division of NAL. Currently, he is the Professor Emeritus in MS Ramaiah Institute of Technology in the Dept of Instrumentation Technology. His current research activities include modelling, identification, multi-sensor data fusion, fuzzy systems, genetic algorithms, and neural networks. He has authored a book on Modelling and Parameter Estimation of Dynamic System, published by IEE, UK, in 2004. He has 100 publications to his credit.