Fuzzy Logic based Adaptive Kalman and H infinity Filtering Schemes

Reshma, V.[1], Girija G.[2], Raol, J. R.[3]

[1]Asst. Professor, Dept. of Electronics and Communications, MSRIT, and Research Scholar, Jain University, Bangalore.
[2]Sr. Scientist (former), CSIR-NAL, Bangalore.
[3]Prof. Emeritus, MSRIT, Bangalore.

ABSTRACT

Some adaptive Kalman filtering schemes are first studied and fuzzy logic based approaches are evaluated for target tracking applications. Then, fuzzy logic based adaptation schemes are presented for an H infinity based filtering algorithm, and evaluated in the sensor data fusion application. The schemes are illustrated for simple and/or maneuvering target tracking applications with numerical examples implemented in MATLAB. Performance metrics and plots are presented.

Keywords: Adaptive filtering, Kalman filter, fuzzy logic, H infinity filter, maneuvering target, sensor data fusion.

INTRODUCTION

Tracking of a moving object/target is a process of estimating the positions and other relevant information with applications as: i) determination of satellite orbit, ii) tracking simple and maneuvering targets, and iii) robotics, in the latter either a robot needs a tracking algorithm for its path and motion planning, and/or a robot itself is tracked by its user from a monitoring/control center. Usually, Kalman filter (KF) is used for estimation of the states of such an object using noisy measurements, and this provides an effective means of estimating the states of the system when it is well defined and the noise statistics are known [1]. However, in real life situations, often these noise statistics are not completely known. The optimal linear KF, for its successful application requires a good knowledge of process and measurement noise statistics, i.e. KF is (theoretically) optimal only if the system dynamics, and noise statistics are known accurately, in fact correctly, and in that case interestingly the KF is a conditionally optimal filter, since its performance depends on correct or good choice of these parameters. Depending on the application, uncertainty in any of these information can lead to filter divergence. In such cases, it is necessary to use some adaptive mechanisms in a KF. Here, we compare a few methods of adaptive tuning of KF: i) heuristic approach, ii) optimal estimation method, and iii) two fuzzy logic (FL) based methods [2]. Performance of these methods is evaluated using simulated target tracking data and some performance metrics. Then, we present FL based approaches to adaptively tune an H infinity (HI) filter, in the deterministic domain, that is an equivalent filter to KF. This FL-augmented/tuned HI filter is evaluated in a sensor data fusion application with simulated data. We find that in all the three, somewhat different applications considered and evaluated independently, FL based approach performs better than the other approaches.

KALMAN FILTER

KF essentially utilizes i) mathematical models of the dynamic system, described by difference or differential equations (in the state space form), ii) actual, and mostly noisy measurements of the dynamic systems, and iii) and the weighted sum of predicted state and measured data (→ residuals) to generate optimal estimates of the states [3]. It has an algorithmic/recursive structure that is amenable to digital computer implementation. For the sake of brevity we only give equations for the discrete time KF. The state space model of a dynamic system in discrete domain is expressed by

$$x(k+1) = \varphi x(k) + Gw(k) \tag{1}$$

$$z(k) = Hx(k) + v(k) \tag{2}$$

In (1), x is the state of the system, and w is a white Gaussian process noise sequence with zero mean and covariance matrix Q; and in (2), z is the observation vector, and v is a white Gaussian measurement noise sequence with zero mean and covariance matrix R; ϕ is the state transition matrix, and H is the measurement model. Using the known model of the dynamic system, statistics Q and R of the noise processes, and noisy measurements z(.), properly tuned KF obtains the optimal estimates of the system states x. The discrete KF equations are given as

Time propagation/evolution

State estimate $$\tilde{x}(k+1) = \phi\,\hat{x}(k) \quad (3)$$

Covariance (a priori) $$\tilde{P}(k+1) = \phi\hat{P}(k)\phi^T + GQG^T \quad (4)$$

Measurement/data update

Residual/innovations $$e(k+1) = z(k+1) - H\,\tilde{x}(k+1) \quad (5)$$

Kalman Gain $$K = \tilde{P}H^T(H\tilde{P}H^T + R)^{-1} \quad (6)$$

Filtered estimate $$\hat{x}(k+1) = \tilde{x}(k+1) + K\,e(k+1) \quad (7)$$

Covariance (a posteriori) $$\hat{P} = (I - KH)\tilde{P} \quad (8)$$

Kalman gain function/matrix can be also written as

$$K = \tilde{P}H^T S^{-1}\ ;\ S = H\tilde{P}H^T + R \quad (9)$$

In (9), S is the theoretical/predicted covariance matrix of the residuals. The actual residuals can be computed from the measurement data update cycle by using (5), and their standard deviations (or absolute values) can be compared with the standard deviations obtained by taking the square roots of the diagonal elements of S, i.e. from (9). Any mismatch between these two quantities indicates that the performance of the KF is not satisfactory, since the filter is not tuned properly.

HEURISTIC METHOD

As we can easily see that the KF performance is dependent on the comparative magnitudes of the Q and R matrices, rather than on their (fixed) absolute magnitudes, this can be seen by combining (4) and (9), and this aspect can be specified as a ratio defined as norm (Q)/norm(R). This property of the noise statistics can be employed to take a practical advantage in real applications where a good estimate of the measurement noise covariance R can be obtained from manufacturer's sensor specifications and laboratory calibrations; then, using this value of R, the Q can be approximated by selecting a suitable proportionality associated to R using a heuristic method/approach (HM). Assuming a constant R value, the simplest form for Q would be via a constant proportionality factor q_1 so that we have [2]

$$Q = q_1 R \quad (10)$$

In (10), q_1 could be determined using the data from extensive experiments on the system and based on analysis and engineering judgment. For a typical target tracking application based on analysis of a series of flight test data of the target collected by distributed sensors, the following form for Q can be used [2]

$$Q_k = \left[q_1\sqrt{R_k}\exp(-q_2 k\Delta t)\right]^2; \quad k = 1, 2, \ldots, N \quad (11)$$

In (11), as $k \to N$, the term $\exp(-q_2 k\Delta t) \to$ small value and the effect of Q on the estimation reduces. It would be necessary to tune only the two values of q_1 and q_2. One may arrive at a different form for Q_k depending on the application and analysis. The HM is computationally simple and might work well for many applications. But, it requires studies with large amount of post flight/operations/experimental data.

OPTIMAL STATE ESTIMATE BASED METHOD

This method is based on the aim to optimally improve the state estimation performance, under the conditions discussed in the introduction [3]. Assuming a steady performance over the most recent N_w samples or sampling times (a sliding

International Journal of Enhanced Research in Science, Technology & Engineering
ISSN: 2319-7463, Vol. 5 Issue 7, July-2016

window), unique estimate of K and R_m can be obtained even if a unique estimate of Q is not obtained. If matrix $\hat{S}$ is chosen as one of the parameters to be estimated, then an estimate of $\hat{S}$ is obtained using (see (9) also)

$$\hat{S} = \frac{1}{Nw}\sum_{k=i-Nw+1}^{i} e(k)e^{T}(k) \tag{12}$$

In (12), e(k) are the residuals. Once (12) is computed, the following equations [3] are utilized in the given sequence to determine the optimal Q (assuming that matric G is invertible)

$$\begin{aligned} \tilde{P}_c &= K\hat{S}(H^T)^{-1} \\ \hat{P}_c &= (I-KH)\tilde{P}_c \\ \hat{Q} &= G^{-1}(\tilde{P}_c - \phi\hat{P}_c\phi^T)G^{-T} \end{aligned} \tag{13}$$

FUZZY LOGIC BASED APPROACHES

The FL approach is based on the principle of covariance matching, i.e. the matching of the theoretical and actual covariance matrices/values [4,5]. Here, the estimates of actual residual covariance are computed from using the filtered residuals as in (12), and the theoretical/prediction values, (9) (as provided by the KF) are compared and the covariance of process noise Q is tuned until these two agree in some sense (say norm, or individual diagonal terms). FL is then used to implement the covariance matching method for adaptation; utilizing this technique, Q and R matrices can be adaptively tuned with a fuzzy inference system (FIS). FL allows for a degree of uncertainty and gradation as against the crisp logic that is based on yes (say, 1) or no (say, 0). FL is a multivalued logic, and the characteristic function is generalized to take an infinite number of values between 0 and 1: e.g. triangular form as a (fuzzy) membership function (MF) as shown in Figure 1 which also depicts a typical FIS.

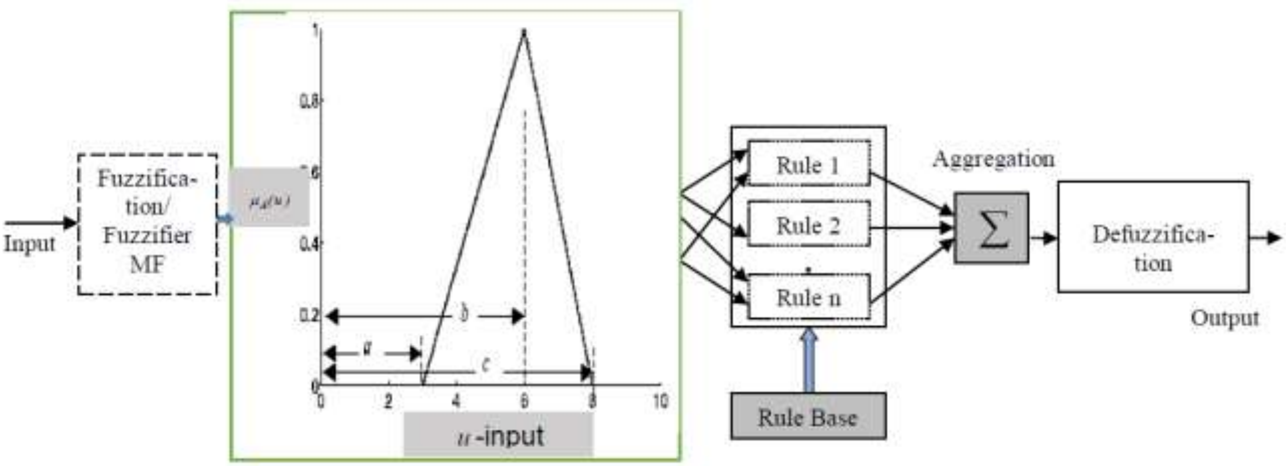

Figure 1 FIS with fuzzy membership function (triangular)

The constants in the MF define the shape and size of MF, and y-axis represents the membership value/grade of belonging of the input variable u, to the fuzzy set. FL can model any continuous linear or nonlinear function or system, and it is 'If…, Then' rule based system, Figure 1. In a FL based system, each input partially fires all rules in parallel and the system acts as an associative processor as it computes the output. The system then combines the partially fired 'Then' part fuzzy sets in a sum and converts this sum to a scalar or vector output, and hence it is called fuzzy associative/additive system, the core of every FIS. The FIS performs an exhaustive search of the rules in the knowledge base/rule base to determine the degree of fit for each rule for a given set of causes. The I/O are crisp variables. The several rules contribute to the final result to varying degrees. Any process that is to be fuzzified, is passed thr' the FIS, and the defuzzified output is then used for further processing: i) predictive analysis, ii) control, and/or iii) estimation. Then any system that uses this process and FIS is called FL based system.

As described in the previous section/s, if there is any discrepancy between the theoretical covariance and the actual statistical analysis of the innovation sequence, FIS can be used to adjust a based on the size of the discrepancy, and Q or R is tuned in such a manner as to reduce this discrepancy using FIS. This approach has the merits: i) it is simple in its formulation, ii) it is able to handle inaccurate information, iii) it is able to include any heuristic/acquired knowledge about the system, and iv) it relaxes some of the a priori assumptions on the noise processes.

FIS for R with known Q

In this case, the adaptation/tuning of R is carried out assuming Q is known; large values of R imply inaccurate measurement data. In such a case, we give less weightage to the measurements and higher weightage (of course relatively) to the prediction in the KF. Thus, we have the theoretical covariance of the innovation given by

$$S(k) = H(k)\widetilde{P}(k)H(k)^T + R(k) \quad (14)$$

The actual/sample covariance of the innovation is computed by using a moving average across a window as in (12). The window size is chosen empirically to give some good smoothing, on the average. The difference between (12) and (14) is used to derive adjustments for R based on its actual value as follows, a variable called degree of matching (DoM) is defined to get the size of the difference

$$\mathrm{DoM}(k) = S(k) - \hat{S}(k) \quad (15)$$

The DoM is used by an FIS to derive R values. From (14) it is clear that increasing R increases S. Hence, R can be varied to reduce the DoM value as [5]

Rule 1: If $\mathrm{DoM}(k) \cong 0$, $S(k)$ and $\hat{S}(k)$ are nearly equal, then maintain R at the same value.
Rule 2: If $\mathrm{DoM}(k) > 0$, means that $S(k) > \hat{S}(k)$, then decrease R.
Rule 3: If $\mathrm{DoM}(k) < 0$, means that $S(k) < \hat{S}(k)$, then increase R.

The adaptation of $R(i,i)$ is equivalent to the adaptation in $\mathrm{DoM}(i,i)$. The FIS generates tuning of R by creating a correction factor $\Delta R(k)$ to be added or subtracted to all the diagonal elements of the R matrix at each instant of time using

$$R(k) = R(k-1) + \Delta R(k) \quad (16)$$

So, it is clear that $\mathrm{DoM}(k)$ is the input to the FIS and $\Delta R(k)$ is the output which is generated sequentially.

FIS for Q with known R

If the large values of Q occur, these imply large uncertainties in the process model and hence, less weightage should be given to the predicted value of the state and more weightage to the measurement data. By incorporating (4) into (14) we can re-write (14) as

$$S(k) = H(k)(\emptyset(k)\hat{P}(k-1)\emptyset(k)^T + Q(k))H(k)^T + R(k) \quad (17)$$

It is assumed that R(k) is known, then it is clear from (17), that if Q increases, S also increases. This mismatch between $S(k)$ and $\hat{S}(k)$ can be used to create a correction term using a similar procedure as for estimating R(k) and adapting the value of Q using the rules [5]

Rule 1: If $\mathrm{DoM}(k) \cong 0$, $S(k)$ and $\hat{S}(k)$ are nearly equal, then maintain Q at the same value.
Rule 2: If $\mathrm{DoM}(k) > 0$, means that $S(k) > \hat{S}(k)$, then decrease Q
Rule 3: If $\mathrm{DoM}(k) < 0$, means that $S(k) < \hat{S}(k)$, then increase Q (18)

For each element in the diagonal matrix Q, an FIS can generate the tuning factor, i.e. $\Delta Q(k)$ is obtained and added/subtracted to correct the Q(k) value for each element in the main diagonal of the matrix Q as

$$Q(k) = Q(k-1) + \Delta Q(k) \quad (19)$$

If there is no direct correspondence between the dimensions of S, Q and DoM, empirical considerations can be used in the FIS to overcome this problem; using the preceding 3 rules, a typical fuzzy system for the input DoM and output ΔR could be formulated as [5]

If DoM is negative medium, then ΔR = Increase Large.
If DoM is negative small, then ΔR = Increase.
If DoM is zero, then ΔR = Maintain.
If DoM is positive small, then ΔR = Decrease.
If DoM is positive medium, then ΔR = Decrease Large. (20)

In FL, the input variable DoM defines the universe of discourse U_{DoM} and the output variables ΔR define the universe of discourse $U_{\Delta R}$. Commonly used membership functions are trapezoidal, triangular, Gaussian or their combination. A suitable defuzzification procedure is used to get the crisp values at each step.

A specific FL approach

Another specific, but similar approach (to one in section 5.2) is a obtained as in (17), but re-written as

$$S(k+1) = H\tilde{P} H^T + R(k+1) = H\{\Phi \hat{P}(k) \Phi^T + \hat{Q}(k)\}H^T + R(k+1) \quad (21)$$

$$= H(\Phi \hat{P}(k) \Phi^T + \sigma^2(k)\bar{Q}(k))H^T + R(k+1) \quad (22)$$

In (22), $\hat{Q}(k) = \sigma^2(k)\bar{Q}(k)$, where $\bar{Q}(k)$ is a fixed a priori known covariance matrix. Then, current Q(.) is altered (in fact $\sigma^2(.)$) at each sampling instant based on, if the innovation : i) is neither too near nor too far from zero, then leave the estimate of Q(k) almost unchanged, ii) is very near to zero, then reduce the estimate of Q(k), and iii) is very far from zero, then increase the estimate of Q(k). This is intuitively appealing since, it achieves the covariance matching. In fact, this mechanism would adjust the value of Q(.) in such a proportion so as to adjust the value of S(.) in (22), and hence, in turn match with the actual covariance of the residual, thereby achieving the covariance matching. This adjustment mechanism can be implemented using FL; at each sampling instant, the input variable (to FL/S-membership function) is given by the parameter

$$r_s(k+1) = \frac{r(k+1)}{\sqrt{s(k+1)}} \quad (23)$$

In (23), r(k+1) is the actual innovation component and s(k+1) is the $(k+1)^{th}$ value of S, then $r_s(k+1)$ gives the measure of relative amplitude of innovation compared to its theoretical assumed value. The following If . . . Then . . . rules can be used to generate output variables; the fuzzy rule based system has r_s as input variables and Ψ as output variables [4], with MFs of r_s and Ψ denoted as m_r and m_Ψ

If r_s is near zero, then Ψ is near zero.
If r_s is small, then Ψ is near one.
If r_s is medium zero, then Ψ is a little larger than one.
If r_s is moderately large, then Ψ is moderately larger than one.
If r_s is large, then Ψ is large. (24)

EVALUATION OF THE METHODS

We evaluate the preceding adaptive filtering schemes using three illustrative cases.

Non-maneuvering target

In the first case, the moving target position data are considered for the three axis-x,y,z frame of reference using the state and measurement models having the form

$$x(k+1) = \varphi x(k) + Gw(k) \quad (25)$$

$$z(k) = H x(k) + v(k) \quad (26)$$

The state vector x consists of target position (p), velocity (v) and acceleration (a) in each of the axes, x, y and z. The basic state transition matrix, process noise matrix and observation matrix for each axis that can be used for generating the simulated data are

Transition matrix $\varphi = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}$ (27)

Process noise gain matrix G= $\begin{bmatrix} \frac{T^2}{2} \\ T \\ 1 \end{bmatrix}$ (28)

Measurement matrix H = [1 0 0] (29)

The state vector x = [x_p, x_v, x_a, y_p, y_v, y_a, z_p, z_v, z_a], thus, for more states the models of (27)-(29) are put in a block diagram matrix for simulation. It is to be noted that (p, v, a) used as subscripts indicate the position, velocity and acceleration respectively. Process noise with σ= 0.001 (standard deviation) is added to the true data to generate the actual state trajectories. A low value of process noise variance yields nearly a constant acceleration motion. The noise variances in each of the coordinate axes are assumed equal. Position measurements in all the three axes are generated by addition of measurement noise with σ = 3.33. Measurements are generated for a duration of 100 sec. with T = 0.25 sec. An interactive GUI (graphical user interface) based MATLAB code is written for generating the simulated data as well the results of the adaptive KF schemes. It also computes the mean of residuals, percentage fit error (PFE), root mean square error position (RMSP) and root mean square error velocity (RMSV) errors. Table 1 gives the PFE for these four adaptive filtering methods: i) heuristic (HMQ), ii) optimal (OSQ), iii) FL based (FLQ), and iv) DoM (FL) based. The numerical values indicate that the performance of all the four adaptive filtering schemes is almost similar in terms of fit errors, however, it can be seen that the PFE from the fuzzy logic based method (FLQ) are lower. Figure 2 depicts the RSS (root sum square) position errors for the four methods. It is seen from Table 1 that the FLQ based method holds a good promise for adaptive KF, and should be further explored for other filtering methods.

Table 1 Position (%) fit errors for AKFs (Case 1)

Tuning methods	x	y	z
HMQ (Heuristic)	0.9256	0.3674	1.6038
OSQ (Optimal)	0.9749	0.3873	1.6895
FLQ (FL based)	**0.8460**	**0.3358**	**1.4659**
DoM (FL based)	0.9840	0.3912	1.7046

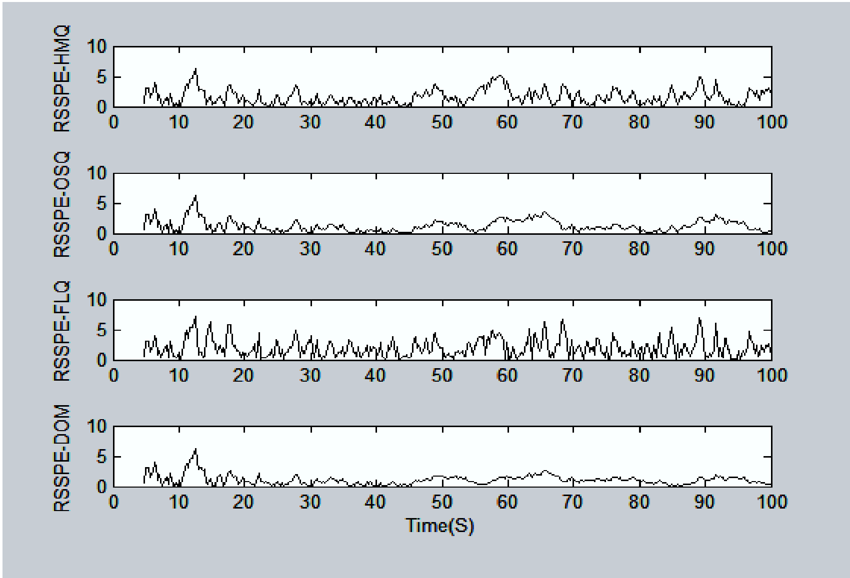

Figure 2 RSS position errors for the four adaptive KFs (Case 1).

Maneuvering target

For the second case we have generated the target motion data in the Cartesian x-y frame of reference of a target moving with constant velocity which undergoes a coordinated turn during a portion of its trajectory [6]. The state vector x consists of target position and velocity in the x and y axes. The simulated data of a target moving in x-y plane with 4 states $[x, v_x, y, v_y]$ are generated using initial conditions [0 0 10 1] for a period of 250 sec. with a sampling interval of 0.5 sec. The target is assumed moving with constant velocity till 24.5 sec., undergoes a coordinated turn during 25.0 to 50 sec. and continues with its constant velocity motion till 250 sec. The state and measurement models are given as

a) constant velocity model

$$\begin{bmatrix} x(k+1) \\ v_x(k+1) \\ y(k+1) \\ v_y(k+1) \end{bmatrix} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ v_x(k) \\ y(k) \\ v_y(k) \end{bmatrix} + \begin{bmatrix} T^2/2 & 0 \\ 0 & T \\ T^2/2 & 0 \\ 0 & T \end{bmatrix} w(k) \; ; \; Y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ v_x(k) \\ y(k) \\ v_y(k) \end{bmatrix} + v(k) \qquad (30)$$

b) Coordinated turn model

For this model, the target is moving in the x-y plane with constant speed and turning with a constant angular rate. The position and velocity evolve along circular arcs. The circular motion can be described by the discrete state model with the same measurement model as equation (30)

$$\begin{bmatrix} x(k+1) \\ v_x(k+1) \\ y(k+1) \\ v_y(k+1) \end{bmatrix} = \begin{bmatrix} 1 & \frac{\sin(\omega T)}{\omega} & 0 & -\frac{1-\cos(\omega T)}{\omega} \\ 0 & \cos(\omega T) & 0 & -\sin(\omega T) \\ 0 & \frac{1-\cos(\omega T)}{\omega} & 1 & \frac{\sin(\omega T)}{\omega} \\ 0 & \sin(\omega T) & 0 & \cos(\omega T) \end{bmatrix} \begin{bmatrix} x(k) \\ v_x(k) \\ y(k) \\ v_y(k) \end{bmatrix} + w(k) \qquad (31)$$

For simulation (the coordinated turn model), a turn rate of -5 deg./sec. and a process noise covariance of 1 are considered and while the target is in constant velocity motion the process noise is assumed to have a variance of 0.03. Measurement data of position along x and y axes are generated by adding measurement noise with covariance of 100. The known value of measurement noise covariance, R=100 is used in all the cases. The target motion is assumed decoupled in the two axes in the adaptive KFs implemented and a constant velocity state model is used in the filters for estimation and the noisy measurements of position are used for measurement update. Initial conditions for the filter are chosen as $\hat{x}_0 = [0.2 \;\; 0.01 \; 10.5 \; 1.05]$.

The tuning factors used in the three filters for this case of simulated data are: i) q_1=0.009 for HMQ, ii) window length N=20 for MLQ, and iii) low=0, high= 1 for FL based method (FLQ). Table 2 gives the PFE metric values for the three methods. Figure 3 shows the estimated position states x, estimated velocity states and estimation errors compared with the measurement errors, using FL based method. It is clear from that the constant velocity model is able to adapt to the coordinated turn model fairly well. The delay in the velocity estimate compared to the true state could be reduced by having faster sampling of the data. Similar results were observed for all the 3 adaptive filters considered. Figure 4 shows the autocorrelation functions (ACRs), and the RMS position errors from the three adaptive schemes. Figure 5 shows the norm of the estimation error covariance matrix.

Table 2 PFE for the three adaptive KF (AKF) schemes (Case 2)

Method	PFE-x Position	PFE-x Velocity	PFE-y Position	PFE-y Velocity
FLQ	**5.0096**	**1.272**	**5.041**	**1.233**
OSQ	5.309	1.502	5.326	1.507
HMQ	5.070	1.297	5.081	1.259

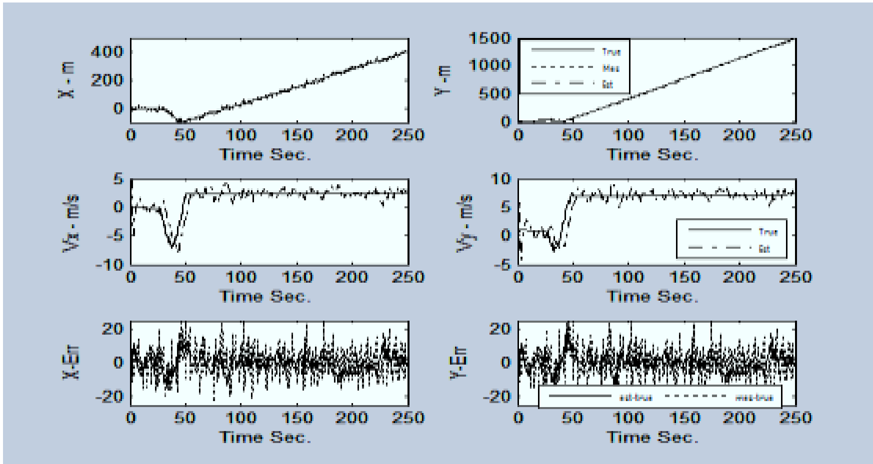

Figure 3 Position, velocity state estimates & position errors: FLQ AKFs (Case 2).

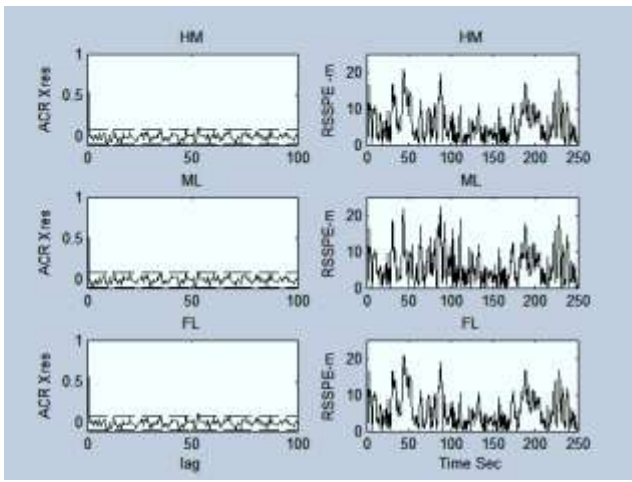

Figure 4 ACRs & RSS position errors: HM-, ML- Q, and FLQ for AKFs (Case 2).

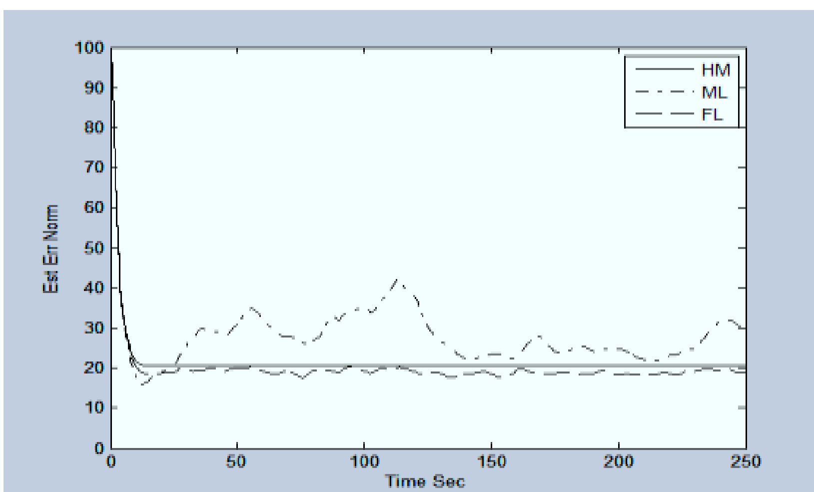

Figure 5 Estimation error covariance norms: HM-, ML- and FLQ adaptive KFs (Case 2).

Once again from case 2, it is found that the FLQ scheme yields somewhat better performance than other two adaptive estimation schemes. One can notice that the numerical PFE values in Table 2 (second case) are somewhat larger than the corresponding values in Table 1 (first case), because for the second case higher values of the (co-)variances are used. This also ascertains the logical performance of the filtering schemes. As such all the three schemes perform well even though they do not carry the coordinated turn model in the KF (however, the simulated data were generated by using the coordinated turn model), but utilize only the constant velocity model, the reason is that this modelling uncertainty, by a large extent is accounted/compensated by the process noise and that too in an adaptive way.

Tracking and data fusion using fuzzy logic augmented H infinity filter

In the third case we consider fuzzy logic augmented H-Infinity (HI) filter for object tracking. An algorithm that employs FL rules is used to adapt an HI filter for target tracking application in a two-sensor data fusion scenario. The sensor data fusion is a process wherein data/information from more than one sensor is fused using some arithmetic, logical, and/or probabilistic formula/rule or method to obtain enhanced information and reduced uncertainty in the prediction. The sensor data fusion has evolved as an independent discipline and has risen to a very high level of sophistication at all data fusion levels: i) kinematic, ii) image, and iii) decision. In the present case for the target model we consider 2 DoF (degrees of freedom) model as

An object is tracked with HI filter associated with each sensor-channel. The object's state vector has two components: position and velocity. The measurement model for each sensor is given by

With m=1,2 (number of sensors). The HI filtering problem differs from KF in two respects: i) the white noise processes w(.) and v(.) are replaced by unknown and yet deterministic (non-random) disturbance of finite energy, and ii) a pre-

$$x(k+1) = \varphi x(k) + G w(k)$$

$$z^m(k) = H x(k) + v^m(k)$$

$$\varphi = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}; \quad G = \begin{bmatrix} T^2/2 \\ T \end{bmatrix}$$

$$E\{v^m(k)\} = 0; \quad Var\{v^m(k)\} = R_v^m$$

specified positive real number (gamma, a scalar parameter, in general γ^2) is defined. The aim of the filter is to ensure that the energy gain (in terms of HI norm) from the disturbances (and initial state error energy) to the estimation error, i.e. the state error is less than this number. This number can be called a threshold for the magnitude of the transfer function between estimation error and the input disturbance energies. From the robustness point of view we see that the H-infinity concept, at least in theory, would yield a robust filtering algorithm. The sensor locations/stations employ individual HI filter to create two sets of track files. The performance is evaluated in terms of state errors using simulated data. The estimates are obtained for each sensor (i=1,2) using HI a posteriori filter as described. The covariance matrix (known in case of HI filter as Gramian) time propagation is given as [7]

$$P_i(k+1) = \varphi P_i(k)\varphi^T + GQG^T - \varphi P_i(k)[H_i^T \;\; L_i^T] R_i^{-1} \begin{bmatrix} H_i \\ L_i \end{bmatrix} P_i(k)\varphi^T$$

$$R_i = \begin{bmatrix} I & o \\ 0 & -\gamma^2 I \end{bmatrix} + \begin{bmatrix} H_i \\ L_i \end{bmatrix} P_i(k)[H_i^T \;\; L_i^T]$$

$$K_i = P_i(k+1)\, H_i^T\, (I + H_i P_i(k+1) H_i^T)^{-1}$$

The HI filter gain is given as

The measurement update of states is obtained by

$$\hat{x}_i(k+1) = \varphi \hat{x}_i(k) + K_i(z_i(k+1) - H_i \varphi \hat{x}_i(k))$$

The fusion of the estimates from the two sensors by SVF (state vector fusion) can be obtained by

$$\hat{x}_f(k+1) = \hat{x}_1(k+1) + \hat{P}_1(k+1)(\hat{P}_1(k+1) + \hat{P}_2(k+1))^{-1}(\hat{x}_2(k+1) - \hat{x}_1(k+1))$$

$$\hat{P}_f(k+1) = \hat{P}_1(k+1) - \hat{P}_1(k+1)(\hat{P}_1(k+1) + \hat{P}_2(k+1))^{-1}\hat{P}_1^T(k+1)$$

The fused state vector and the fused covariance (Gramian) of the fused state utilize the individual estimate state vectors (of each sensor) and covariance matrices.

In HI filter, γ is the tuning parameter and in the present case it is tuned using FL/FIS. The γ can be considered to be proportional to the magnitude/s of the noise processes/s. As we increase γ we tell the filter that the state is likely to change more at each time step, this results in a filter that is more responsive to changes in the measurement. This can be viewed as an increase in the bandwidth of the filter. Values of γ that are too small result in slow convergence of the optimization algorithm, and possibly convergence to a local minimum that is larger than that achieved by more appropriate values of γ. Values of γ that are too large would cause an oversensitivity of the algorithm to local gradient, and might result in divergence. FL is accommodated in tuning of the gamma parameter. We consider four approaches: i) the HI a posteriori filter, ii) FLHI 1, iii) FLHI 2, and iii) FLHI 3.

In FLHI 1, the trapezoidal MF is used to decide the value of gamma. Iteration is done along the x axis. The value obtained from this MF is added to 1 to get the value of gamma. The MF has positive slope for first few iterations as value of gamma should be increased gradually in the beginning. Then, the slope becomes 0 for most of the iterations, since we maintain the higher gamma value. In the last few iterations, slope can be made negative since high gamma value is not very significant.

In FLHI 2, we use the fact that the residual error behavior with respect to iterations is a damped sinusoidal wave. Hence, if the error (residual) and change in error (difference between the current residual and the last residual, i.e. error rate) is given, we can find whether the iteration is in the beginning or final stages. In this method, it is done using two sigmoidal functions, one each for residual and the change in residuals. Sigmoidal function for residuals makes sure that if error is less, corresponding output is high and if error is more, corresponding output is low. The output is always between 0 and 1. Sigmoidal function for change in residual works in a similar way. If one value (either residual or change in residual) is low and other is high, it means that process is still in the initial iterations. In such conditions, the gamma value should be low. Hence, ‘min’ operation is performed between the two membership functions

For FLHI 3, we use; a) FIS consisting of antecedent (input functions), consequent (output functions) and fuzzy rules, b) Mamdani type FIS, c) the centroid method for defuzziffication, and d) the properties of damping sinusoidal function and gamma value to form fuzzy rules. In any FIS, fuzzy implication provides mapping between input and output fuzzy sets. Basically, a fuzzy If…Then… rule is interpreted as a fuzzy implication. The antecedent membership functions that define the fuzzy values for input (residual error and change of error) and for consequent output. The rules for the inference in FIS are created based on the past experiences and intuitions. Three rules are used to tune the parameter gamma

Rule 1: IF residual value is high (irrespective of change in residual value), THEN gamma value should be low. (This rule is created based on the fact that, when error is high; gamma value should be less since process is in the initial stage).

Rule 2: IF change in residual value is high (irrespective of residual value), THEN gamma value should be low. (This rule is created based on the fact that, when change in error is high; gamma value should be less since process is still in the initial iteration).

Rule 3: IF residual is low AND change in residual is low, THEN value of gamma is high. (This rule is created based on the fact that only when both error and change in error are low, the value of gamma should be high).

General observations for the rules are: i) If the error (residue) is high, the output function is at a higher value. Hence, the area of aggregate of all output functions will have more area towards one. If the error is low, the output function is at a lower value and the output function of this rule does not contribute much to the aggregate output function. This rule pulls the gamma value towards 1; ii) If the change in error is high, the output function is at a higher value. Hence, the area of aggregate of all output functions will have more area towards 1. If the change in error is low, the output function is at lower value and the output function of this rule does not contribute much to the aggregate output function. This rule pulls the gamma value towards 1; and iii) For both error and change in error, a min operation is performed between the two functions. Hence the output is high only if both the parameters (residue and change in residue) are low.

The target data are generated using constant acceleration model with process noise increment. With sampling interval T = 0.1 s, a total of N = 500 scans are generated. The normalized random noise is added to the state vector and the measurements of each sensor are corrupted with random noise. The sensor could have dissimilar measurement noise variances (Sensor 2 having higher variance than Sensor 1).The initial conditions for the state vector are x(0)=[200 0.5]. The performance of the fusion by both the methods i.e. HI filter (HIF) and FL-based HI filter (FLHI) is evaluated in terms of estimation energy gain, and this can be considered an upper bound on the maximum energy gain from input to the output. The PFEs and the % state errors are given in Table 3. From the performance plots shown in Figures 6-8 (for FL based HI), and Table 3, it is seen that the FLHI based approaches give better results than only HI filter.

Table 3 PFE for HI & FLHI filters (Case 3)

Method/Filter	HI norm (fused) & PFE (S1 and S2)	Sensor 1 and 2 F- fused	% SE position SE-State error	% SE velocity
HI	0.0522; 0.4428; 0.5684	S1	0.2100	5.5567
		S2	0.2723	6.3979
		F	0.1740	5.4370
FLHI-1	0.0211; 0.4034; 0.5184	S1	0.1264	4.3979
		S2	0.1612	5.8195
		F	0.1028	4.5666
FLHI-2	**0.0081; 0.3847; 0.4960**	S1	**0.0799**	**3.2170**
		S2	**0.1000**	**4.7149**
		F	**0.0637**	**2.9672**
FLHI-3	0.0084;0.3848;0.4949	S1	0.0786	3.2434
		S2	0.0975	4.8234
		S3	0.0639	3.2156

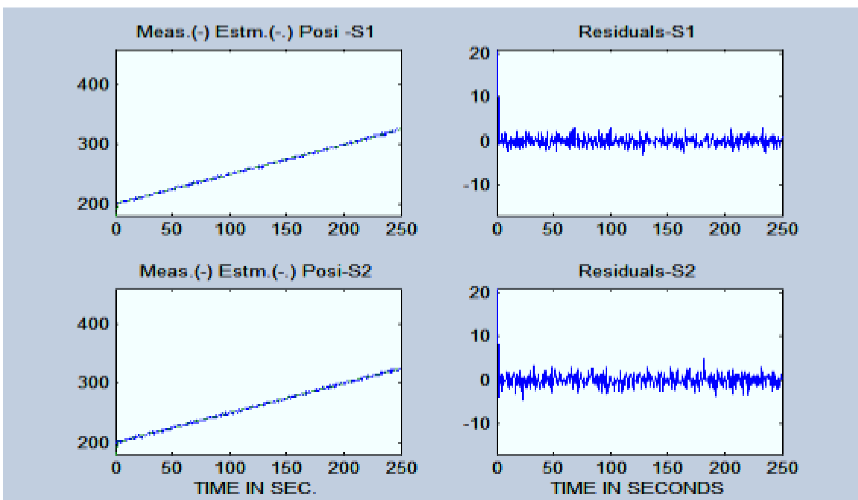

Figure 6 Measurements and residuals (Case 3)

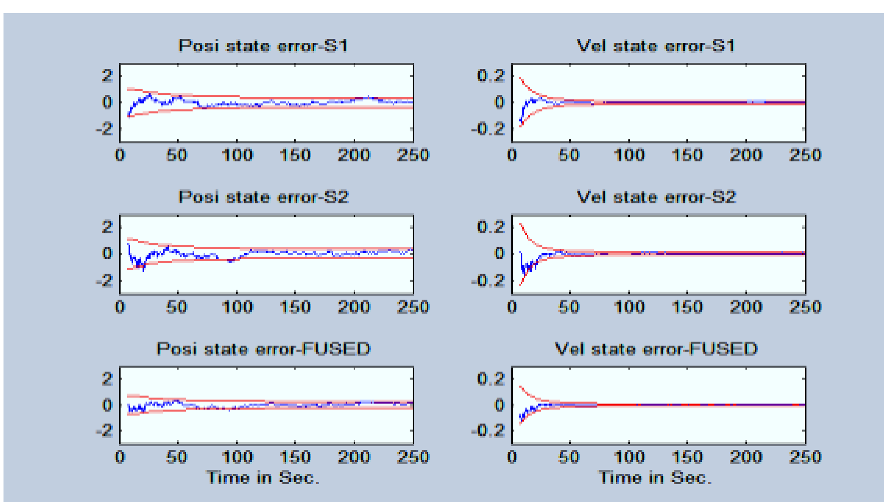

Figure 7 State-errors with bounds (Case 3)

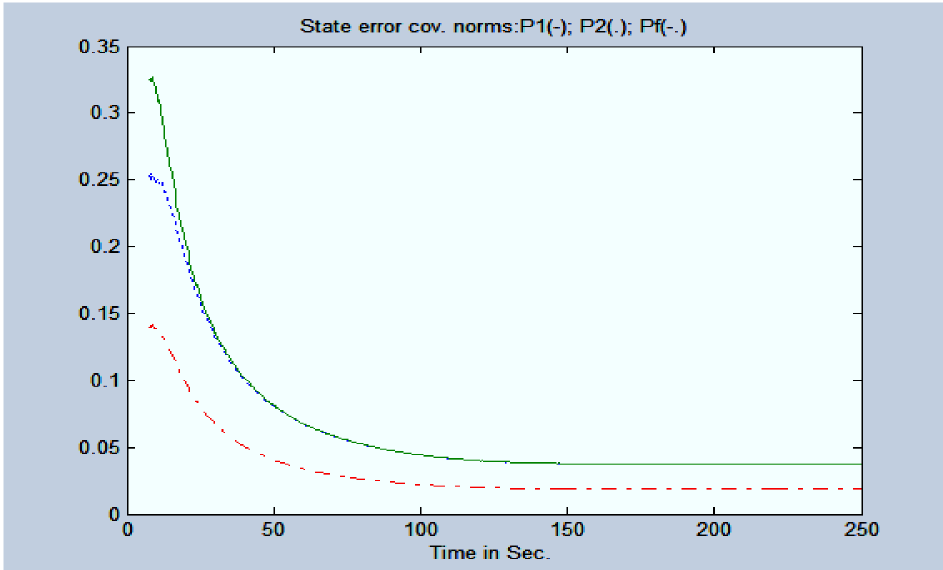

Figure 8 Norms of covariance matrices (Case 3)

CONCLUSIONS

We have discussed four adaptive schemes that can be used in conjunction with either KF or HI filter, and have evaluated their performances for one/or more of the realistically generated simulated data in the target tracking applications. Also, we have presented FL based adaptive scheme for H infinity filter in tracking and sensor data fusion scenario. From the results of the three cases studied, we infer that the FL based adaptive KF and HI filter perform better than non-fuzzy logic based schemes.

ACKNOWLEDGMENT

The first author is grateful to Alva, A., Advith J., Akash R. S, and Varun K. R., the project team for help in MATLAB based implementations for Case 3.

REFERENCES

[1]. Bar-Shalom, Y., X.-Rong Li, and T. Kirubarajan. Estimation with Applications to Tracking and Navigation, John Wiley & Sons Inc., New York, USA, 2001.
[2]. Girija, G. and J. R. Raol. Evaluation of adaptive Kalman filtering methods for target tracking applications, Paper AIAA 2001-4106; doi:10.2514/6.2001-4106, AIAA Guidance, Navigation, and Control Conference and Exhibit, Montreal, Canada, 2001.
[3]. Maybeck, P. S. Stochastic Models, Estimation, and Control. Vol. 1, Academic Press, New York, 1979.
[4]. Jetto, L., Longhi, S., and Vitali, D. Localization of a wheeled mobile robot by sensor data fusion based on fuzzy logic adapted Kalman filter, Control Engg. Practice, Vol. 4, pp. 763-771, 1999.
[5]. Escamilla, P. J., and N. Mort, Development of a fuzzy logic-based adaptive Kalman filter, Proc. of the European Control Conference ECC'01, Porto, Portugal, pp. 1768-1773. September 4-7, 2001.
[6]. Naidu, V. P. S., G. Girija, and N. Santhakumar. Three model IMM-EKF for tracking targets executing evasive maneuvers. AIAA-66928, 45th AIAA conference on Aerospace Sciences, Reno, USA, 2007.
[7]. B. Hassibi, A.H. Sayed and T. Kailath. Linear estimation in Krein spaces - Part II: Applications, IEEE Trans Automatic Control, Vol. 41, No. 1, pp 34-49, Jan 1996.