Multisensor Data Fusion Architecture Based on Adaptive Kalman Filters and Fuzzy Logic Performance Assessment

P. J. Escamilla-Ambrosio
Department of Automatic Control
and Systems Engineering,
University of Sheffield, Sheffield, UK.
COP99PJE@Sheffield.ac.uk

N. Mort
Department of Automatic Control
and Systems Engineering,
University of Sheffield, Sheffield, UK.
N.Mort@sheffield.ac.uk

Abstract - *In this work a novel Multi-Sensor Data Fusion (MSDF) architecture is presented. First, each measurement-vector coming from each sensor is fed to a Fuzzy Logic-based Adaptive Kalman Filter (FL-AKF); thus there are N sensors and N FL-AKFs working in parallel. The adaptation in each FL-AKF is in the sense of dynamically tuning the measurement noise covariance matrix R employing a fuzzy inference system (FIS) based on a covariance matching technique. Second, another FIS, here called a fuzzy logic assessor (FLA), is monitoring and assessing the performance of each FL-AKF. The FLA assigns a degree of confidence, a number on the interval [0, 1], to each one of the FL-AKF outputs. Finally, a defuzzification scheme obtains the fused state-vector estimate based on the confidence values. The effectiveness and accuracy of this approach is demonstrated in a simulated example. Two defuzzification methods are explored and compared; results show good performance of the proposed approach.*

Keywords: Multisensor data fusion, adaptive Kalman filtering, fuzzy logic, performance assessment.

1 Introduction

The Multi-Sensor Data Fusion (MSDF) approach is described as the acquisition, processing, and synergistic combination of information gathered by various knowledge sources and sensors to provide a better understanding of a phenomenon under consideration [1]. Different MSDF techniques have been explored recently. These techniques vary from those based on well-established Kalman filtering methods [2], [3], to those based on recent ideas from soft computing technology [4], [5]. However, little work has been done in exploring architectures that consider the combination of both these approaches. In this work a novel MSDF architecture that combines these approaches is explored. This architecture is built integrating the fuzzy logic-based adaptive Kalman filter developed recently by Escamilla and Mort [6] and a fuzzy logic performance assessment scheme.

The general idea explored in this approach is the combination of the advantages that both Kalman filtering and fuzzy logic techniques have. On the one hand, Kalman filtering is recognized as one of the most powerful traditional techniques of estimation: the Kalman filter provides an unbiased and optimal estimate of a state-vector in the sense of minimum error variance [7]. On the other hand, the main advantages derived from the use of fuzzy logic techniques, with respect to traditional schemes, are the simplicity of the approach, the capability of fuzzy systems to deal with imprecise information, and the possibility of including heuristic knowledge about the phenomenon under consideration.

The remainder of this paper is organized as follows. In section 2, after describing the traditional Kalman filtering approach, the fuzzy logic-based adaptive Kalman filter (FL-AKF) is summarized. Section 3 describes the proposed MSDF architecture. The effectiveness of this approach is demonstrated in a simulated example outlined in section 4. Finally, in section 5 the conclusions and perspectives of this work are given.

2 Formulation of the fuzzy logic-based adaptive Kalman filter

2.1 The traditional Kalman filter

Given a discrete-time controlled process described by the linear stochastic difference equations:

$$x_{k+1} = A_k x_k + B_k u_k + w_k \tag{1}$$

$$z_k = H_k x_k + v_k \tag{2}$$

where k represents the discrete-time index, x_k is the system state-vector, u_k is the input vector, z_k is the measurement-vector, w_k and v_k are uncorrelated zero-mean Gaussian white noise sequences with covariance matrices Q_k and R_k respectively; the Kalman filter algorithm can be described by next group of equations [9],

a) Time update (or prediction) equations:

$$\hat{x}^-_{k+1} = A_k \hat{x}_k + B_k u_k \tag{3}$$

$$P_{k+1}^{-} = A_k P_k A_k^T + Q_k \quad (4).$$

b) Measurement update (or correction) equations:

$$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1} \quad (5)$$

$$\hat{x}_k = \hat{x}_k^- + K_k [z_k - H_k \hat{x}_k^-] \quad (6)$$

$$P_k = [I - K_k H_K] P_k^- \quad (7),$$

where $\hat{x}_k$ represents the estimate of the system state-vector x_k, P_k is the state-estimate error covariance matrix, and K_k is commonly referred to as the filter gain or the Kalman gain matrix.

Equations (3) and (4) project, from time step k to step k+1, the current state and error covariance estimates to obtain the *a priori* (indicated by the super minus) estimates for the next time step. While equations (5) to (7) incorporate a new measurement into the *a priori* estimate to obtain an improved *a posteriori* estimate.

The term $H_k \hat{x}_k^-$ in (6) is the one-stage predicted measurement $\hat{z}_k$, and the difference $(z_k - H_k \hat{x}_k^-)$ is referred to as the innovation sequence or residual, generally denoted as r and defined by

$$r_k = (z_k - H_k \hat{x}_k^-) \quad (8).$$

The Kalman filter algorithm starts with initial conditions at $k = 0$, being $\hat{x}_0^-$ and P_0^-. With the progression of time, as new measurements z_k become available, the cycle estimation-correction of states and the corresponding error covariances can follow recursively ad infinitum.

2.2 The fuzzy logic-based adaptive Kalman filter

As described previously, the traditional Kalman filter formulation assumes complete *a priori* knowledge of the process and measurement noise covariance matrices, Q_k and R_k. However, in most practical applications these matrices are initially estimated or, in fact, are unknown. The problem here is that the optimality of the estimation algorithm in the Kalman filter setting is closely connected to the quality of the *a priori* noise statistics [10]. It has been shown how poor estimates of the input noise statistics may seriously degrade the Kalman filter performance, and even provoke the divergence of the filter [11], [12]. From this point of view it can be expected that an adaptive formulation of the Kalman filter will result in a better performance or will prevent filter divergence.

In this section, an on-line fuzzy logic-based adaptive Kalman filter (FL-AKF) is presented [6]. The adaptation is in the sense of using a Fuzzy Inference System (FIS) to dynamically adjust the measurement noise covariance matrix R_k from data as they are obtained. This relaxes the a priori measurement noise statistical assumptions and significantly benefits the Kalman filter states estimates if the measurement noise under it operates change or evolves with time. The main advantages derived from the use of a fuzzy technique, with respect to traditional adaptation schemes, are the simplicity of the approach and the possibility of including heuristic knowledge about the phenomenon under consideration.

The measurement noise covariance matrix R_k represents the accuracy of the measurement instrument, meaning a larger R_k for measured data implies that we trust this data less and take more account of the prediction. Assuming that the noise covariance matrix Q_k is known, here a FIS based on the technique known as covariance-matching [13] has been derived to dynamically adjust the covariance matrix R_k.

The basic idea behind the covariance-matching technique is to make the residuals consistent with their theoretical covariance [10], [14]. In the FL-AKF this is done in three steps; first, having available the innovation sequence or residual r_k its theoretical covariance is calculated as,

$$S_k = H_k P_k^- H_k^T + R_k \quad (9),$$

in the Kalman filter algorithm. Second, the actual covariance $\hat{C}r_k$ of r_k is approximated through averaging inside a moving estimation window [14] of size M,

$$\hat{C}r_k = \frac{1}{M} \sum_{i=i_0}^{k} r_i r_i^T \quad (10),$$

where $i_0 = k - M + 1$ is the first sample inside the estimation window. This means that only the last M samples of r_k are used to estimate its covariance. The window size is chosen empirically to give some statistical smoothing. Third, if it is found that the actual value of the covariance of r_k has a discrepancy with its theoretical value, then a FIS derives adjustments for R_k based on the knowledge of the size of this discrepancy. The objective of these adjustments is to correct this mismatch as well as possible. In order to detect the size of the discrepancy between S_k and $\hat{C}r_k$ a new variable called the Degree of Matching (*DoM*) is defined as,

$$DoM_k = S_k - \hat{C}r_k \quad (11).$$

The main idea of adaptation used by a FIS to dynamically tuning R_k is as follows. It can be noted from (9) that an increment in R_k will increment S_k, and vice versa. This means that R_k can be used to vary S_k in accordance with the value of DoM_k in order to reduce the discrepancies between S_k and $\hat{C}r_k$. From here three general rules of adaptation are defined as:

1. If $DoM_k \cong 0$ (this means S_k and $\hat{C}r_k$ match almost perfectly) then maintain R_k unchanged.

2. If $DoM_k > 0$ (this means S_k is greater than its actual value $\hat{C}r_k$) then decrease R_k.

3. If $DoM_k < 0$ (this means S_k is smaller than its actual value $\hat{C}r_k$) then increase R_k.

Note that the matrices $\hat{C}r_k$, S_k, R_k and DoM_k are all of the same size, thus the adaptation of the (i, i) element of R_k can be made in accordance with the (i, i) element of DoM_k; i=1,2,…,m; m=size of z_k. Thus, a single-input-single-output (SISO) FIS is used to sequentially generate the tuning or correction factors for the elements in the main diagonal of R_k, and this correction is made in this way,

$$R_k(i,i) = R_{k-1}(i,i) + \Delta R_k \quad (12),$$

where ΔR_k is the tuning factor that is added or subtracted from the element (i,i) of R_k at each instant of time. ΔR_k is the FIS output and $DoM_k(i,i)$ is the FIS input. A graphical representation of this adjusting process is shown in Figure 1.

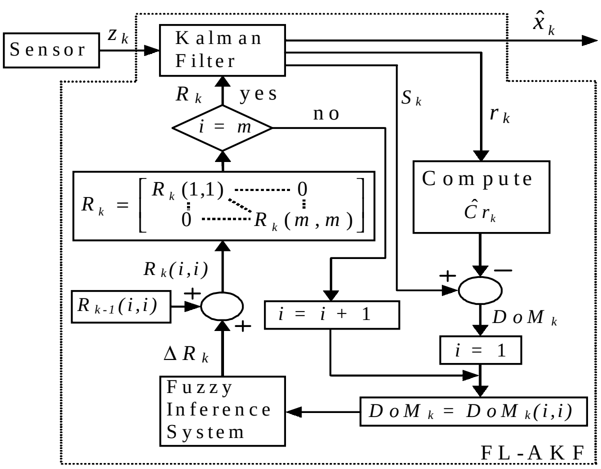

Figure 1. Graphical Representation of the adjusting process of R_k.

Following the general rules of adaptation, the FIS can be implemented considering three fuzzy sets for DoM_k: N = Negative, ZE = Zero, and P = Positive; and three fuzzy sets for ΔR_k: I = Increase, M = Maintain, and D = Decrease. These membership functions are shown in Figure 2. There, the parameters that define the fuzzy sets can be changed in accordance with the system under consideration. Hence, only three fuzzy rules are included in the FIS rule base:

1. If $DoM_k = N$, then $\Delta R_k = I$
2. If $DoM_k = ZE$, then $\Delta R_k = M$
3. If $DoM_k = P$, then $\Delta R_k = D$.

Thus, using the compositional rule of inference sum-prod and the center of area (COA) defuzzification method, R_k is adjusted in each FL-AKF as given in (12). From experimentation it was found that a good size for the moving window in (10) is M = 15.

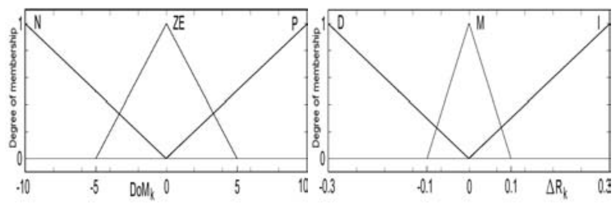

Figure 2. Membership functions for DoM_k and ΔR_k.

2.3 Fault detection and recovery algorithm.

Additionally to the adaptation procedure, the FL-AKF has been provided with a fault detection and recovery algorithm. First a variable called Residual Compatibility rC_k is defined as,

$$rC_k^i = \frac{\left| r_k^i \right|}{\sqrt{S_k(i,i)}} \quad (13).$$

The value of rC_k^i gives a measure of the actual amplitude of the i-th element of the residual compared with its theoretical value $S_k(i,i)$ (the corresponding element in the main diagonal of S_k) at each instant of time k. This value is around 1 if both quantities are similar, but it increases abruptly if a transient fault, i. e., a blunder, is present in the measured data. Thus, if the value of rC_k^i is greater or equal than a threshold (α) then a transient fault is declared. If the value of rC_k^i remains greater than α for more that an instant of time, then a persistent fault, i. e. a cycle slip, is declared. The algorithm shown in Figure 3 is implemented in order to detect and compensate the occurrence of faults. This algorithm assigns the value of zero to the i-th element of r_k if a transient fault is detected (through the value of its corresponding rC_k^i). This is the best guess that can be made due to the zero average characteristic of the residual. However if a persistent fault is detected, this does not works any more, in that case r_k^i is assigned with a random number selected from a zero mean Gaussian white noise sequence with covariance $S_k(i,i)$; which in presence of a persistent faulty data is a consistent guess. The variables $T\text{-}Fault_k$ and P_Fault_k are used to register the occurrence of transient and persistent faults (1 = fault present, 0 = not fault present), respectively. In this approach α is selected as 4. The above fault detection and recovery algorithm is evaluated after equation (5) and before equations (6) and (10) in the FL-AKF algorithm.

if $rC_k^i \geq \alpha$ then
    T-Fault$_k$ = 1
    $r_k^i = 0$
    if T-Fault$_{k-1}$ = 1 then
        P_Fault$_k$ = 1
        $r_k^i = \sqrt{S_k(i,i)} * randn$
    end
end

Figure 3. Fault detection and recovery algorithm.

3 Description of the proposed MSDF Architecture

Under the assumption that the system in consideration is completely observable, in this section a novel Multi-Sensor Data Fusion (MSDF) architecture is presented. The objective of the proposed architecture is to combine the measurement-vectors coming from N disparate sensors, each one with different measurement dynamics and noise characteristics, to obtain a fused state-vector estimate that better reflects the actual value of the parameters being measured. To reach this objective, first each measurement-vector coming from each sensor is fed to a FL-AKF; thus there are N sensors and N FL-AKFs working in parallel as shown in Figure 4. Second, another FIS, here called a fuzzy logic assessor (FLA), is monitoring and assessing the performance of each FL-AKF. The FLA assigns a degree of confidence-vector, denoted as c_k, to the FL-AKF state-vector estimate. This is made based on the current value of the absolute value of the size of discrepancy between S_k and $\hat{C}r_k$, $|DoM_k|$; and the current value of the noise covariance matrix, R_k. Finally, a defuzzificator obtains the fused state-vector estimate through a defuzzification scheme based on the assigned confidence values. Here different defuzzification procedures can be explored and compared. Figure 4 shows a graphical representation of the proposed MSDF architecture. The FL-AKF was described in section 2. The FLA and the defuzzificator are described in next sections.

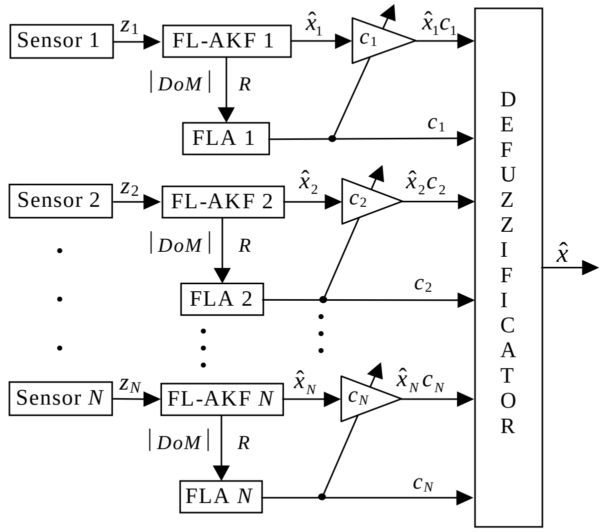

Figure 4. Proposed MSDF architecture.

3.1 The Fuzzy Logic Assessor (FLA)

The FLA assigns a degree of confidence-vector c_k to the FL-AKF state-vector estimate in the following way. The degree of confidence c_k^i for the i-th element ($i = 1,\ldots, n$) in the state-vector estimate is calculated in a recursively way based on the (i,i) corresponding elements in $|DoM_k|$ and R_k. A graphical representation of this process is shown in Figure 5. The degree of confidence c_k^i, a number on the interval [0, 1], is an indicator of the level in which each FL-AKF state estimate reflects the true value of the parameter being measured. At the same time, the degree of confidence acts as a weighting factor that tells a defuzzificator at what confidence level it should take each FL-AKF state estimate.

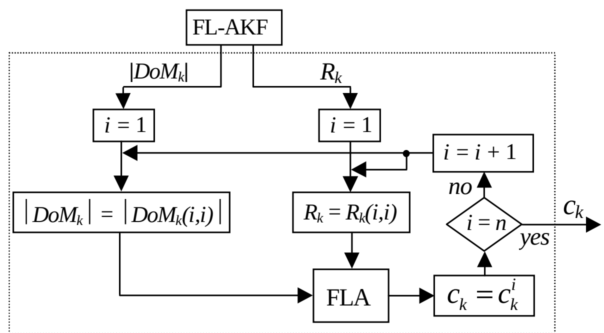

Figure 5. Process of calculating the degrees of confidence.

Each FLA is implemented using two inputs: the current values of $|DoM_k|$ and R_k; and one output, the degree of confidence c_k (see Figure 5). The membership functions for $|DoM_k|$ and R_k are shown in Figure 6. There the fuzzy labels mean: ZE = Zero, S = Small, and L = Large. For the output c_k, three fuzzy singletons are defined with the labels: G=1=Good, AV=0.5=Average, and P=0=Poor. Here as well, the parameters that define the fuzzy sets can be changed in accordance with the application under consideration. Thus, nine rules complete the fuzzy rule base of each FLA, as given in Table 1 known as a decision table. The fuzzy rules are based on two simple heuristic considerations. First, if the value of $|DoM_k|$ is near to zero and the value of R_k is near to zero, then it means the filter is working almost perfectly; thus the degree of confidence assigned by the FLA is near the maximum, 1. Second, if one or both of these values increases far from zero, it means that the filter performance is degrading; thus the degree of confidence assigned by the FLA is decreased in accordance, up to the minimum 0. Thus, using the compositional rule of inference sum-prod and the center of area defuzzification method, the FLA assigns a degree of confidence to the state-estimates made by the FL-AKF.

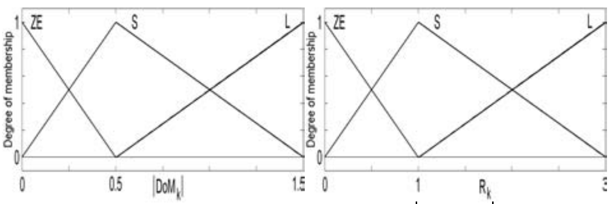

Figure 6. Membership functions for $|DoM_k|$ and R_k.

Table 1 Decision Table for the FLA rule base

R / $\|DoM\|$	ZE	S	L
ZE	G	G	AV
S	G	AV	P
L	AV	P	P

3.2 The defuzzificator

The defuzzificator obtains the fused state-vector estimate through a defuzzification scheme based on the assigned confidence values. Different defuzzification methods can be explored to select the best one for a particular application. The methods proposed here are the center of area (COA) and a variation of the maximum, the winner takes all (WTA). In the COA method the i-th element $\hat{x}_k^i$ of the fused state-vector estimate $\hat{x}_k$, at instant of time k, is obtained as,

$$\hat{x}_k^i = \frac{\sum_{j=1}^{N} \hat{x}_k^{i(j)} c_k^{i(j)}}{\sum_{j=1}^{N} c_k^{i(j)}}; \quad \hat{x}_k = \begin{bmatrix} \hat{x}_k^1 \\ \vdots \\ \hat{x}_k^n \end{bmatrix} \qquad (14)$$

where $\hat{x}_k^i$ is the i-th element in the fused state-vector estimate, $i = 1,\ldots, n$; $\hat{x}_k^{i(j)}$ is the i-th element in the state-vector estimate obtained by the j-th FL-AKF, $j = 1,\ldots, N$; N = number of sensors; $c_k^{i(j)}$ is the i-th degree of confidence assigned to the i-th element in the state-vector estimate obtained by the j-th FL-AKF (see Figure 5); k denotes the instant of time. Thus the first part of Equation (14) is processed recursively n times in order to obtain the final fused state-vector estimate $\hat{x}_k$ ($\hat{x}_k$ is an $n \times 1$ vector).

In the WTA method the fused state-vector estimate $\hat{x}_k$ is obtained as,

$$\hat{x}_k^i = \arg\max_j \left(c_k^{i(j)}\right); \quad \hat{x}_k = \begin{bmatrix} \hat{x}_k^1 \\ \vdots \\ \hat{x}_k^n \end{bmatrix} \qquad (15),$$

where the function $\arg\max_j(\cdot)$ returns as output the j-th FL-AKF state estimate $\hat{x}_k^{i(j)}$ ($j = 1,\ldots, N$) which has the maximum degree of confidence $c_k^{i(j)}$ at each instant of time k. Thus the first part of Equation (15) is processed recursively n times in order to obtain the final fused state-vector estimate $\hat{x}_k$.

In order to prevent possible conflicts, one modification in each method was incorporated. For the COA method, if the sum of all the degrees of confidence for the i-th element in the state-vector estimate is equal to zero, then the fused output is simply the average of the N elements. For the WTA, if there is more than one maximum degree of confidence for the i-th element in the state-vector estimate, then the element with the first maximum encountered is given as the fused estimate.

4 Illustrative example

In this section an example with three noisy and faulty sensors is outlined to demonstrate the effectiveness and accuracy of the proposed MSDF architecture.

Consider the following linear system, which is a modified version of a tracking model [15], [16],

$$\begin{bmatrix} x_{k+1}^1 \\ x_{k+1}^2 \\ x_{k+1}^3 \end{bmatrix} = \begin{bmatrix} 0.77 & 0.20 & 0.00 \\ 0.25 & 0.75 & 0.25 \\ 0.05 & 0.00 & 0.75 \end{bmatrix} \begin{bmatrix} x_k^1 \\ x_k^2 \\ x_k^3 \end{bmatrix} + \begin{bmatrix} w_k^1 \\ w_k^2 \\ w_k^3 \end{bmatrix} \qquad (16)$$

$$\begin{bmatrix} z_k^{1(j)} \\ z_k^{2(j)} \\ z_k^{3(j)} \end{bmatrix} = H^j \begin{bmatrix} x_k^1 \\ x_k^2 \\ x_k^3 \end{bmatrix} + \begin{bmatrix} v_k^{1(j)} \\ v_k^{2(j)} \\ v_k^{3(j)} \end{bmatrix}; \qquad j = 1, 2, 3 \qquad (17)$$

where H^j is the *j-th* measurement matrix, x_k^1, x_k^2, and x_k^3 are the position, velocity, and acceleration, respectively, of a flying object; $z_k^{1(j)}$, $z_k^{2(j)}$, and $z_k^{3(j)}$ are observations of the object position, velocity and acceleration, made by the *j*-th sensor, respectively; { w_k^i }, i = 1, 2, 3 is an uncorrelated zero-mean Gaussian white noise sequence with matrix covariance $Q = 0.02I_3$; { $v_k^{i(j)}$ } is a noise sequence defined for each sensor as is given in Figures 7 and 8. Initial conditions are $\hat{x}_o^i = 0$, $P_0 = 0.01 I_3$. The actual value of the measurement noise covariance matrices for each sensor is assumed as unknown, but an initial value is given as shown in Figure 7. The measurement matrices and noise profiles for each sensor are given in Figures 7 and 8.

$$H^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad H^2 = \begin{bmatrix} 1.2 & 0 & 0 \\ 0 & 0.8 & 0 \\ 0 & 0 & 1.4 \end{bmatrix}; \; H^3 = \begin{bmatrix} 0.6 & 0 & 0 \\ 0 & 1.2 & 0 \\ 0 & 0 & 0.7 \end{bmatrix}$$

$$\begin{bmatrix} v_k^{1(1)} \\ v_k^{2(1)} \\ v_k^{3(1)} \end{bmatrix} = \begin{bmatrix} noise\ 1 \\ noise\ 2 \\ noise\ 3 \end{bmatrix}; \quad \begin{bmatrix} v_k^{1(2)} \\ v_k^{2(2)} \\ v_k^{3(2)} \end{bmatrix} = \begin{bmatrix} noise\ 2 \\ noise\ 3 \\ noise\ 1 \end{bmatrix}; \quad \begin{bmatrix} v_k^{1(3)} \\ v_k^{2(3)} \\ v_k^{3(3)} \end{bmatrix} = \begin{bmatrix} noise\ 3 \\ noise\ 1 \\ noise\ 2 \end{bmatrix}$$

$$R_0^1 = \begin{bmatrix} 2.5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 10 \end{bmatrix}; \; R_0^2 = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad R_0^3 = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 5 \end{bmatrix}$$

Figure 7. Measurement matrices, noise profiles, and initial measurement noise covariance matrices for each sensor.

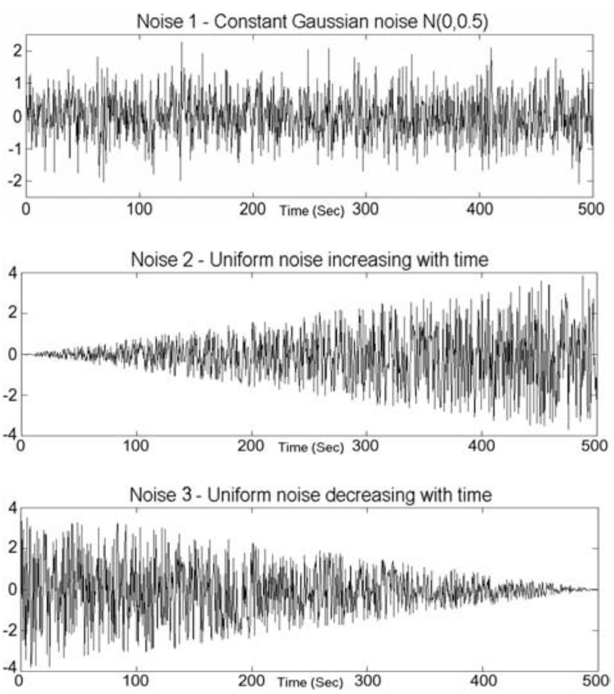

Figure 8. Noise profiles in the measurements.

MATLAB code was developed to simulate the process described by Equation (16) and sensors described by Equation (17) together with the proposed MSDF architecture considering the fusion of the data coming from the three sensors. Two experiments were carried out. In the first experiment no faults were introduced in the measured data (referred to as non-faulty sensors experiment, NFSE). In the second experiment blunders and cycle slips were introduced in the measured data to simulate the occurrence of faults (referred to as faulty sensors experiment, FSE). Figure 9 shows the time instants, indicated by the subscripts, and the value of the faulty data introduced in each sensor measurement for the FSE. The purpose of the experiments is to observe the performance of the proposed MSDF architecture under both faulty and non-faulty sensors. In each experiment the simulation was carried out for 500s with a sample time of 0.5s. The simulation results are presented in next section.

$$\begin{bmatrix} z^{1(1)}_{100:120,300:400,500:520,700:720} \\ z^{2(1)}_{150,250,350,450} \\ z^{3(1)}_{200,400,600,800} \end{bmatrix} = \begin{bmatrix} -5 \\ -5 \\ -5 \end{bmatrix}$$

$$\begin{bmatrix} z^{1(2)}_{150,350,550,750} \\ z^{2(2)}_{100,300,400,500} \\ z^{3(2)}_{250,450,650,850} \end{bmatrix} = \begin{bmatrix} -5 \\ -5 \\ -5 \end{bmatrix}$$

$$\begin{bmatrix} z^{1(3)}_{250,450,650,850} \\ z^{2(3)}_{150,250,350,550} \\ z^{3(3)}_{300,400,600,800} \end{bmatrix} = \begin{bmatrix} -5 \\ -5 \\ -5 \end{bmatrix}$$

Figure 9. Faults introduced in each sensor for the FSE (the symbol ':' is used to specify a range of values, for example 100:120, means from time instant k=100 to k=120).

4.1 Results

For comparison purposes, the following performance measures were adopted:

$$J^i_{xv} = \sqrt{\frac{1}{n} \sum_{k=1}^{n} (x^i_k a - x^i_k v)^2} \qquad (18)$$

$$J^i_{xe} = \sqrt{\frac{1}{n} \sum_{k=1}^{n} (x^i_k a - x^i_k e)^2} \qquad (19)$$

where $x_k^i a$ is the actual value, $x_k^i v$ is the measured value, and $x_k^i e$ is the estimated value of: i = 1 = position, i = 2 = velocity, and i = 3 = acceleration of the flying object at instant of time k, respectively; n = No. of samples.

Table 2 shows the performance measures obtained in the NFSE for each individual FL-AKF and those obtained from the fusion of the three sensors using the proposed

MSDF architecture with both COA and WTA defuzzification methods. Table 3 shows the same performance measures but now for the FSE. Analyzing the data given in Table 2, it is noted that the best state-vector estimate is obtained with the MSDF architecture employing the WTA defuzzification method. Additionally, this state-vector estimate is more exact than the obtained by any of the individual FL-AKFs. The state-vector estimate obtained with the MSDF architecture employing the COA defuzzification method is slightly less accurate than the obtained with the MSDF architecture employing the WTA defuzzification method. The improvement on the position and velocity estimates obtained using both MSDF cases is of around the 5% to 10% with respect to the estimates done by individual FL-AKFs. However, the acceleration estimates are improved only in around the 2%. Figure 10 shows the actual and fused estimated position and its corresponding error obtained with the MSDF architecture using the WTA defuzzification method in the NFSE. Figure 11(a) shows the degrees of confidence assigned to each FL-AKF position estimate and Figure 11 (b) shows the number of the selected FL-AKF which position estimate is given as the fused output by using the WTA defuzzification method in the NFSE.

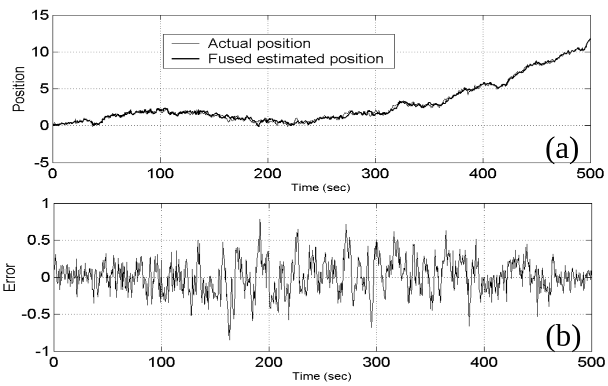

Figure 10. (a) Actual and fused estimated position obtained with the proposed MSDF architecture using the WTA defuzzification method. (b) Corresponding error.

Table 2 Performance measures for sensors without faults

No. of Sensor	J^1_{xv}	J^2_{xv}	J^3_{xv}	J^1_{xe}	J^2_{xe}	J^3_{xe}
1	0.7285	1.2816	1.2964	0.2520	0.2700	0.1943
2	1.4768	1.6359	0.7886	0.2426	0.2686	0.1819
3	1.5496	1.1916	1.3198	0.2468	0.2426	0.2006
Fused COA				0.2257	0.2380	0.1820
Fused WTA				0.2235	0.2386	0.1788

Table 3 Performance measures for sensors with faults

No. of Sensor	J^1_{xv}	J^2_{xv}	J^3_{xv}	J^1_{xe}	J^2_{xe}	J^3_{xe}
1	2.6456	1.3587	1.3345	0.2651	0.2805	0.1950
2	1.5466	1.6875	0.8667	0.2423	0.2681	0.1824
3	1.6355	1.2692	1.3886	0.2471	0.2436	0.2008
Fused COA				0.2298	0.2397	0.1824
Fused WTA				0.2301	0.2410	0.1794

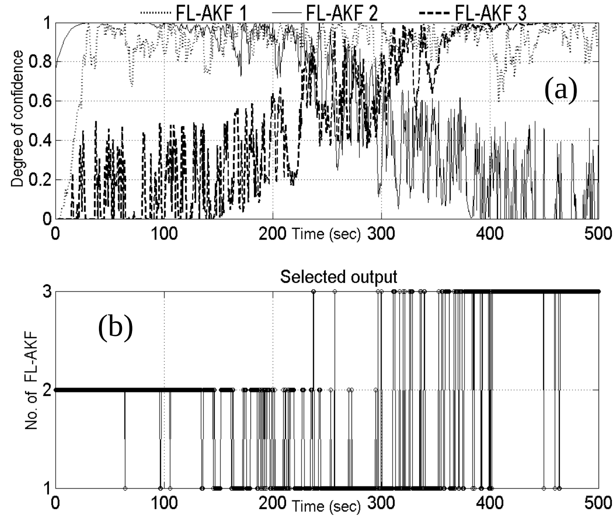

Figure 11. (a) Degrees of confidence assigned to each FL-AKF position estimate. (b) No. of selected FL-AKF.

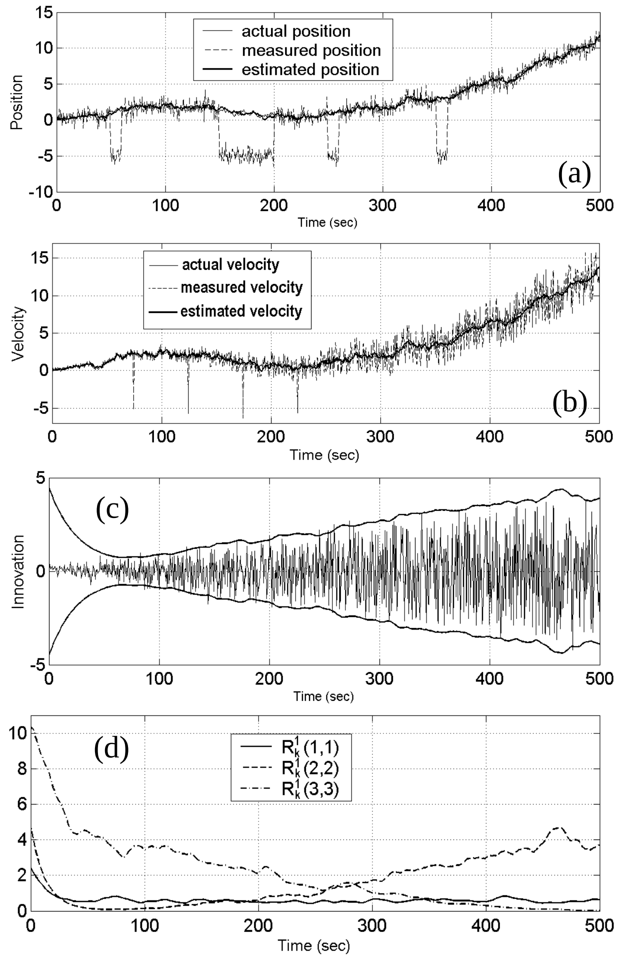

Figure 12. (a) Actual, measured and estimated position - FL-AKF 1 - FSE. (b) Actual, measured an estimated velocity - FL-AKF 1 - FSE. (c) r_k^2 and its corresponding ± theoretical 2σ limits - FL-AKF 1 - FSE. (d) Elements in the main diagonal of R_k^1 - FL-AKF 1 - FSE.

Analyzing the data given in Table 3 it can be noted that the effects caused by the faults introduced on the measured data are eliminated by the fault detection and recovery algorithm incorporated to each FL-AKF. This fault tolerant characteristic can be appreciated in figures 12(a) and 12(b) where the estimated position and velocity, obtained by the FL-AKF 1, in the FSE, are shown. It is remarkable to note that the performance is only slightly degraded with respect to that observed in the NFSE. It means that each FL-AKF is fault tolerant and due to this characteristic a negligible degradation on the fused state estimates obtained by the MSDF architecture is observed, using whatever of the two defuzzification methods. However, note in Table 3 that the performance of the MSDF architecture using the WTA defuzzification method is more affected when faulty data is present. Finally, in order to show the dynamic tuning characteristic of the covariance matrix R in the FL-AKF approach, Figure 12(c) shows the residual corresponding to the estimated velocity obtained with the FL-AKF 1 during the FSE. Additionally, Figure 12(d) shows the main elements in the measurement noise covariance matrix R_k^1 obtained with the FL-AKF 1 during the FSE. Note how both are adjusted in accordance with the noise profile in the measured data.

5 Conclusions

A novel MSDF architecture integrating Kalman filtering and fuzzy logic techniques has been presented. This approach exploits the advantages that both techniques have: the optimality of the Kalman filter and the capability of fuzzy systems to deal with imprecise information using 'common sense' rules. In this approach the linear estimations obtained by individual Kalman filters are improved through dynamically tuning the measurement noise covariance matrix R_k by means of a FIS. This prevents filter divergence and relaxes the a priori assumption about the initial value of R. It is particularly relevant that only three rules are needed to carry out the adaptation. Additionally, in accordance with the results obtained in the illustrated example, the incorporation of the fault detection and recovery algorithm in the FL-AKF practically eliminates the effects of faults, doing the complete system fault tolerant.

The role of the FLA in the proposed MSDF architecture is of great importance because the fusion of the information is carried out based on the degrees of confidence generated on this component no matter what defuzzification method is used. Another important point is that only two variables are needed to monitor the performance of each FL-AKF and only nine 'common sense' rules are used in the FLA rule base.

The results obtained in the illustrative example are promising. They show that the proposed MSDF architecture is effective in situations where there are several sensors measuring the same parameters, but each one has different measurement dynamic and a dynamic noise characteristic is present. Thus the general idea of exploring the combination of traditional together with non-traditional techniques for designing MSDF architectures appears to be a good avenue of investigation.

The choice of the fuzzy sets used in the fuzzy systems was carried out using a trial and error scheme. Obviously this process is time consuming and depends on the problem under consideration. In order to tackle this problem the authors are exploring the idea of using a neuro-fuzzy system to adjust automatically these fuzzy sets. For the case of the fuzzy rules, the general guidelines given for both FL-AKF and FLA showed its effectiveness in the chosen example.

References

[1] Varshney, P. K., *Multisensor data fusion*, Electronics and Communication Engin. Journal, Vol. 9, No. 6, pp. 245-253, 1997.

[2] Luo, R. C., *Multisensor integration and fusion in intelligent systems*, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 19, No. 5, pp. 901-931, 1989.

[3] Gao, Y., E. J. Krakiwsky, M. A. Abousalem, and J. F. McLellan, *Comparison and analysis of centralized, decentralized, and federated filters*, Navigation: Journal of The Institute of Navigation, Vol. 40, No. 1, pp. 69-86, 1993.

[4] Kuo, R. J. and P. H. Cohen, *Multi-sensor integration for on-line tool wear estimation through radial basis function networks and fuzzy neural networks*, Neural Networks, Vol. 12, pp. 355-370, 1999.

[5] Kobayashi F. et al, *Sensor fusion system using recurrent fuzzy inference*, Journal of Intelligent and Robotic Systems, Vol. 23, pp. 201-216, 1998.

[6] Escamilla-Ambrosio, P. J. and N. Mort, *Development of a fuzzy logic-based adaptive Kalman filter*, Proceedings of the 2001 European Control Conference, Porto, Portugal, Sep. 2001, pp. 1768-1773.

[7] Maybeck, P. S., *Stochastic models estimation and control*, Volume 1, Academic Press, New York, 1979.

[8] Welch, G. and G. Bishop, *An Introduction to the Kalman Filter*, The University of North Carolina at Chapel Hill, TR95-045, 1995.

[9] Brown, R. G. and P. Y. C. Hwang, *Introduction to random signals and applied Kalman filtering whit Matlab exercises and solutions*, Third edition, John Wiley & Sons, New York, 1997.

[10] Mehra, R. K., *On the identification of variances and adaptive kalman filtering*, IEEE Trans. Automatic Control, Vol. AC-15, No. 2, pp. 175- 184, 1970.

[11] Fitzgerald, R. J., *Divergence of the Kalman filter*, IEEE Trans. Automatic Control, Vol. AC-16, No. 6, pp. 736- 747. 1971.

[12] Sangsuk-Iam, S. and T. Bullock, *Analysis of discrete-time Kalman filtering under incorrect noise covariances*, IEEE Trans. on Automatic Control, Vol. 35, No. 12, pp. 1304-1308, Dec. 1990.

[13] Mehra, R. K., *Approaches to adaptive filtering*, IEEE Trans. Automatic Control, Vol. AC-17, pp. 693- 698, Oct. 1972.

[14] Mohamed, A. H. and K. P. Shwarz, *Adaptive kalman filtering for INS/GPS*, Journal of Geodesy, Vol. 73, No. 4, pp. 193-203, 1999.

[15] Paik, B.S. and J.H. Oh, *Gain fusion algorithm for decentralised parallel Kalman filters*, IEE Proc. Control Theory Appl., Vol. 147, No. 1, pp. 97-103, 2000.

[16] Chen, G. and C. K. Chui, *A modified adaptive Kalman filter for real-time applications*, IEEE Trans. Aerospace and Electronic Systems, Vol. 27, No. 1, pp. 149-154, 1991.