

Tracking filter and multi-sensor data fusion

G GIRIJA¹, J R RAOL¹, R APPAVU RAJ² and SUDESH KASHYAP¹

¹System Identification Laboratory, Flight Mechanics and Control Division, National Aerospace Laboratories, Bangalore 560 017, India

²Interim Test Range, Chandipur 756 025, India

e-mail: jrraol@css.cmmacs.ernet.in

Abstract. In this paper factorization filtering, fusion filtering strategy and related algorithms are presented. Some results of implementation and validation using realistic data are given.

Keywords. Multi-sensor data fusion; Kalman filtering; tracking.

1. Introduction

Multi-sensor data fusion (MSDF) is defined as the process of integrating information from multiple sources to produce the most specific and comprehensive unified data about an entity, activity or event. As a technology, MSDF is the integration and application of many traditional disciplines and new areas of engineering including communication and decision theory, uncertainty management, estimation theory, digital signal processing, computer science and artificial intelligence. MSDF is expected to result in robust operational performance, extended spatial coverage, increased confidence, reduced ambiguity, improved detection performance, enhanced spatial resolution, improved system operational reliability and increased dimensionality (Hall 1992). At a basic level the processing operations are dominated by numeric procedures involving linear and nonlinear estimation techniques, pattern recognition processes and various statistical operations. For a flight test range the tracking of the flight vehicle and sensor fusion are of great importance.

In the present paper, U-D factorized Kalman filter, state vector fusion filter and a fusion philosophy are described. The composite scheme is termed UDF based MSDF strategy. The strategy is validated using post flight data.

2. U-D factorization Kalman filter

The problem addressed in this paper is represented by the following set of equations.

State model:

$$x_{j+1} = \phi_{j,j+1}x_j + Gw_j. \quad (1)$$

Measurement model:

$$z_j = Hx_j + v_j. \quad (2)$$

Here, x is the state vector, w is the process noise with zero mean and covariance matrix Q , z is the measurement vector and v is the measurement noise with zero mean and covariance matrix R , all of appropriate dimensions. The transition matrix of the state equation (1) is given as

$$\phi = \begin{bmatrix} 1 & \Delta t & \Delta t^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}. \quad (3)$$

Here, Δt is the sampling interval and $H = [1 \ 0 \ 0]$. State vector x has position, velocity, and acceleration as its components. The tracking problem is related to estimation of the position and velocity of a moving object (based on the available measurements) using Kalman filter. This filter is implemented in the factorized form for the present application. It is given in two parts: Time Propagation and Measurement Update algorithms (Bierman 1977).

Time propagation algorithm

State vector evolution (prediction)

$$\tilde{x}_{j+1} = \phi_{j+1,j} \hat{x}_j. \quad (4)$$

Covariance update

$$\tilde{P}_{j+1} = \phi_j \hat{P}_j \phi_j^T + GQG^T. \quad (5)$$

With $\hat{P} = \hat{U}\hat{D}\hat{U}^T$, the time update factors \tilde{U} and \tilde{D} are obtained through modified Gram-Schmidt orthogonalization process (Bierman 1977). The matrix U is an upper triangular matrix with unit elements on its main diagonal and D is a diagonal matrix. Covariance and gain processing algorithms, operating on U and D factors of state error covariance matrix P , are a technique for implementing "square root filtering" without requiring computation of square roots. The U - D Kalman filtering algorithm is considered efficient, stable and accurate for real-time applications (Bierman 1977).

We define $W = [\phi \hat{U} | G]$; $D = \text{diag}[\hat{D}, Q]$ with $W^T = [w_1, w_2, \dots, w_n]$,

Here, ' T ' denotes transpose of vector/matrix. The U, D factors of $\tilde{P} = WDW^T$ may be computed now. For $j = n, n-1, \dots, 2$, the following equations are recursively evaluated as shown below.

$$\left. \begin{aligned} \tilde{D}_j &= \langle w_j^{(n-j)}, w_j^{(n-j)} \rangle_D, \\ \tilde{U}(i, j) &= \langle w_j^{(n-j)}, w_j^{(n-j)} \rangle_D / \tilde{D}_j, \\ w_i^{n-j+1} &= w_i^{(n-j)} - \tilde{U}(i, j) w_j^{(n-j)}, \\ \tilde{D}_1 &= \langle w_1^{(n-1)}, w_1^{(n-1)} \rangle_D, \end{aligned} \right\} i = 1, \dots, j-1. \quad (6)$$

Here subscript D qualifies the weighted inner product with respect to D .

Measurement update algorithm

The measurement update in Kalman filtering combines a priori estimate \tilde{x} and error covariance \tilde{P} with scalar observation $z = a^T x + v$; $a^T = H$ to construct an updated (filtered state) estimate and covariance as follows:

$$\begin{aligned} K &= \tilde{P}a/\alpha, \\ \hat{x} &= \tilde{x} + K(z - a^T \tilde{x}), \\ \alpha &= a^T \tilde{P}a + r, \\ \hat{P} &= \tilde{P} - Ka\tilde{P}. \end{aligned} \quad (7)$$

Here, r is the measurement noise variance (for scalar data processing). Kalman gain K and updated covariance factors \hat{U} and \hat{D} can be obtained from the following equations.

$$\begin{aligned} f &= \tilde{U}^T a; \quad f^T = (f_1 \dots f_n), \\ v &= \tilde{D}f; \quad v_i = \tilde{d}_i f_i, \quad i = 1, 2, \dots, n, \\ \hat{d}_i &= \tilde{d}_i r / \alpha_i; \quad \alpha_1 = r + v_1 f_1; \quad K_2^T = (v_1 0 \dots 0). \end{aligned} \quad (8)$$

For $j = 2, \dots, n$ recursively the following equations are evaluated:

$$\left. \begin{aligned} \alpha_j &= \alpha_{j-1} + v_j f_j, \\ \hat{d}_j &= \tilde{d}_j \alpha_{j-1} / \alpha_j, \\ \hat{u}_j &= \tilde{u}_j + \lambda_j K_j, \\ K_{j+1} &= K_j + v_j \tilde{u}_j, \end{aligned} \right\} \quad \lambda_j = -f_j / \alpha_{j-1}. \quad (9)$$

where $\tilde{U} = [\tilde{u}_1, \dots, \tilde{u}_n]$, $\hat{U} = [\hat{u}_1, \dots, \hat{u}_n]$ and Kalman gain is given by $K = K_{n+1} / \alpha_n$. Here \tilde{d} is predicted diagonal element, and \hat{d}_j is the updated diagonal element of the D matrix. The U - D filter described above is developed in 'C' language and implemented on a DEC Alpha computer. It has been validated using simulated trajectory data and also real flight data.

3. MSDF strategy

Figure 1 shows a hierarchical MSDF architecture for a typical range equipped with several types of sensors for tracking of a flight vehicle (Girija *et al* 1999). It is proposed to have three sets of fused position (x, y, z) and corresponding velocity state-information to the Decision Support System (DSS). The data from ground based radars are combined to yield one set of fused data 1. The second fused data 2 is generated by fusion of INS (inertial navigation system) and GPS (global positioning system, presently not used) data received at the ground station. The angular data from the telemetry (TM) channels which measure only the azimuth (ϕ) and elevation (θ) are combined using least squares method to generate position information. The azimuth and elevation data from EOT (electro-optical transducers) are also combined in a similar manner to generate the position information. Although the UDF block is shown explicitly in figure 1 for various sensor channels, only one UDF is used and the data are processed sequentially by the same filter. These estimates of state vector i.e. position, velocity, acceleration and the error covariance matrices are the input to fusion process to achieve joint or fused state vector estimate based on multiple sensors.

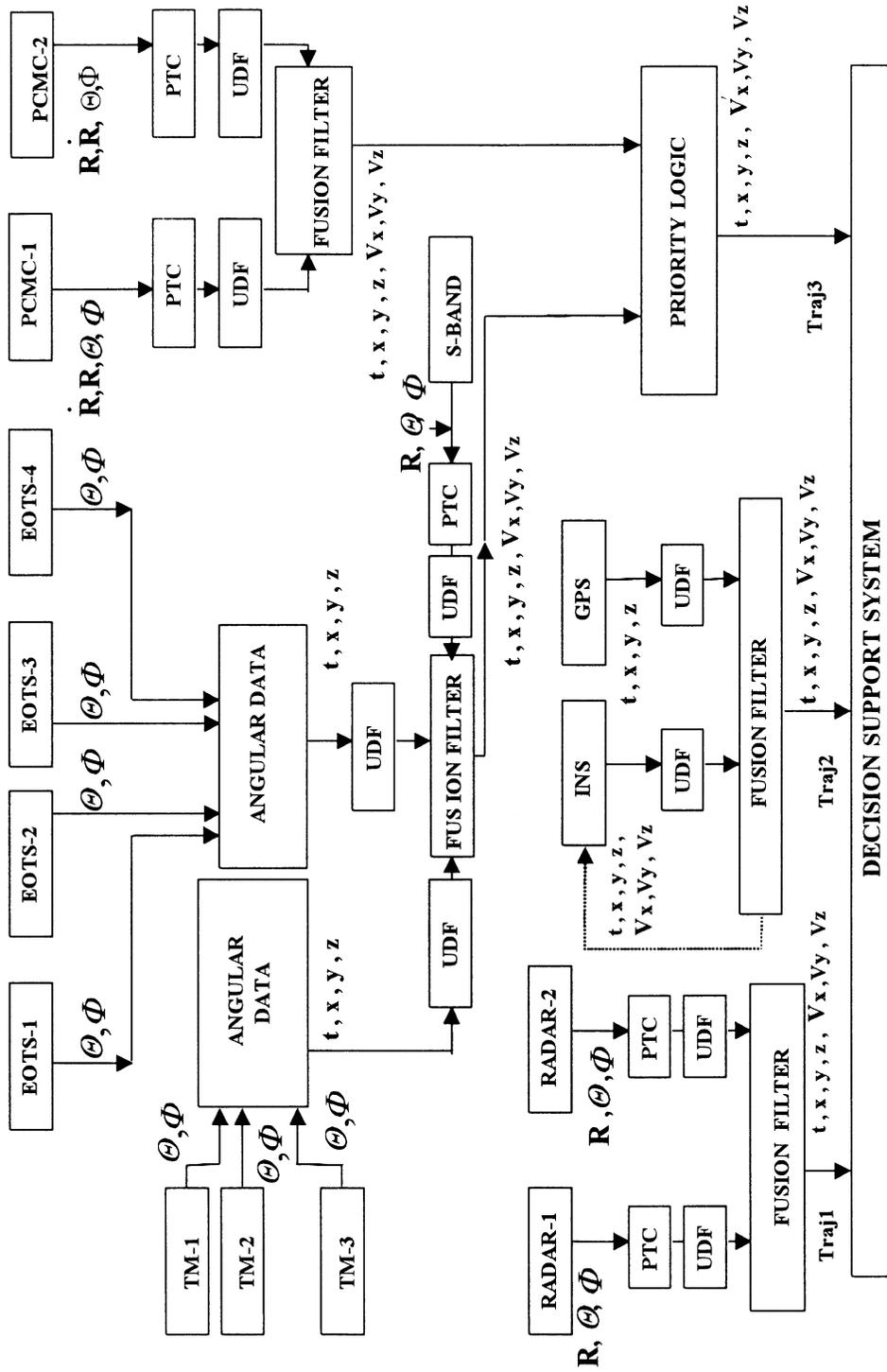


Figure 1. MSDF scheme (PTC-polar to cartesian).

3.1 Fusion algorithm

The fusion algorithm works on the basis of the following equations (Saha 1996):

The fused state vector is given by

$$\hat{x}_F(j/j) = \hat{x}_1(j/j) + \hat{P}_1(j/j)\hat{P}_E^{-1}(j/j)[\hat{x}_2(j/j) - \hat{x}_1(j/j)]. \quad (10)$$

The covariance of the fused state vector is given by

$$\hat{P}_F(j/j) = \hat{P}_1(j/j) - \hat{P}_1(j/j)\hat{P}_E^{-1}(j/j)\hat{P}_1(j/j)^T. \quad (11)$$

The error covariance $\hat{P}_E(j/j)$ given by

$$\hat{P}_E(j/j) = \hat{P}_1(j/j) + \hat{P}_2(j/j)^T. \quad (12)$$

In the present case, the cross covariance between any two-state vectors estimated using data from two sensors is neglected for simplicity. The estimates of the state vectors and the covariance matrices are obtained from the *U-D* filtering process of each channel data and used in fusion filter of equations (10)–(12) directly, e.g.

$$P_1 = U_1 D_1 U_1^T,$$

because no updating of these factors is needed by the fusion filter. The *U-D* filtering algorithm and the fusion filter, (10)–(12), are jointly termed as UDF in this paper.

4. Sensor fusion philosophy

In order to be able to use sensor-channel such as EOT, PCMC, S-Band, two TM, RADAR 1, RADAR 2, INS, and GPS for fusion, it is necessary to develop fusion logic to use the information from these sensors. The fusion logic is mainly based upon priority logic and range limit of individual sensors. Since, for the time being, real data from PCMC, GPS, and 2 telemetry (TM) sensors are not available for simulation and validation, the PCMC and GPS data are replaced by INS data (for the sake of the validation process). Also the INS data is used to generate the synthetic trajectory data from two telemetry sensors using simple triangulation method. The priority logic is decided based on the sensor accuracy within the range of the sensor capability. Based on this priority logic the following sequence could be given to the sensors (of the first module) within the range limit of, say, RL km.

- EOT
- PCMC
- TM and S-Band fusion
- S-Band
- TM
- Track loss

For range more than RL km PCMC radar tracks the target. For the second module which contains the RADARs, the following sequence is followed

- RADAR 1 and RADAR 2 fusion
- RADAR 1
- RADAR 2
- Track loss

For the third module which contains the INS and GPS, the following sequence is followed.

- INS and GPS fusion (GPS data replaced by INS)
- INS
- GPS (GPS data replaced by INS)
- Track loss

Rule base: The rule base for three modules that depends upon the health of the sensors is given. Tables 1–3 cover possible conditions and corresponding actions that can occur in the fusion philosophy.

Table 1. Fusion module 1: For range limit \leq RL (TRAJ 3).

EOT	S-Band	TM	Action 1	Action 2	Action 3
1	1	1	EOT estimation and send to DSS	S-band estimation	TM estimation
1	1	0	EOT estimation and sent to DSS	S-band estimation	TM prediction
1	0	1	EOT estimation and sent to DSS	S-band prediction	TM estimation
1	0	0	EOT estimation and sent to DSS	S-band prediction	TM prediction
0	1	1	EOT prediction	Fusion of TM and S-band Send to DSS	
0	1	0	EOT prediction	Fusion of estimated S-band and predicted TM and send to DSS for $t \leq ms(*)$ $t > ms$	
0	1	0	EOT prediction	S-band estimation and send to DSS	TM prediction
0	0	1	EOT prediction	Fusion of predicted S-band and estimated TM and send to DSS for $t \leq ms$ $t > ms$	
0	0	1	EOT prediction	S-band prediction	TM estimation Send to DSS
0	0	0	If (ivalpcmc \neq 0.0) then PCMC estimation and send to DSS and EOT, S-band, and TM prediction else if (ivalpcmc $=$ 0.0) then track loss , EOT, S-band, and TM prediction and EOT send to DSS		

For range limit $>$ RL

PCMC	ACTION
1	PCMC estimation and send to DSS
0	PCMC prediction and send to DSS

(*) $m = n\Delta t$; n is the no. of sampling intervals.

Table 2. Fusion module 2 (TRAJ 1).

RADAR 1	RADAR 2	Action 1	Action 2
1	1	Fusion and send to DSS	
1	0	Fusion of estimated radar 1 and predicted radar 2	
1	0	RADAR 1 estimation and send to DSS	Predicted RADAR 2
0	1	Fusion of estimated radar 2 and predicted radar 1	
0	1	RADAR 1 prediction	RADAR 2 estimation and send to DSS
0	0	RADAR 1 prediction	RADAR 2 prediction and send to DSS

4.1 Software structure

The software contains the modules and the sub-modules for different fusion channels and fusion levels. The modules are the subroutines of the main program and the sub-modules are subroutines of modules. At present the fusion is done with maximum two similar sensor channels. The fusion software reads the sensor data from the files and performs filtering and fusion process for every fusion module to generate TRAJ1, TRAJ2 and TRAJ3. These trajectories are sent to the DSS for further necessary action.

Fusion module 1

Sub-module 1: Angular telemetry data fusion (triangulation or least square method), filtering and the earth curvature correction if required.

Sub-module 2: S-band co-ordinate conversion to telemetry station. Filtering, fusion and earth curvature correction if required.

Table 3. Fusion module 3 (TRAJ 2)

INS	GPS	Action 1	Action 2
1	1	Fusion and send to DSS	
1	0	Fusion of estimated INS and predicted GPS	
1	0	INS estimation and send to DSS	
0	1	Predicted GPS	
0	1	Fusion of estimated GPS and predicted INS	
0	1	INS prediction	GPS estimation and send to DSS
0	0	INS prediction and send to DSS	GPS prediction

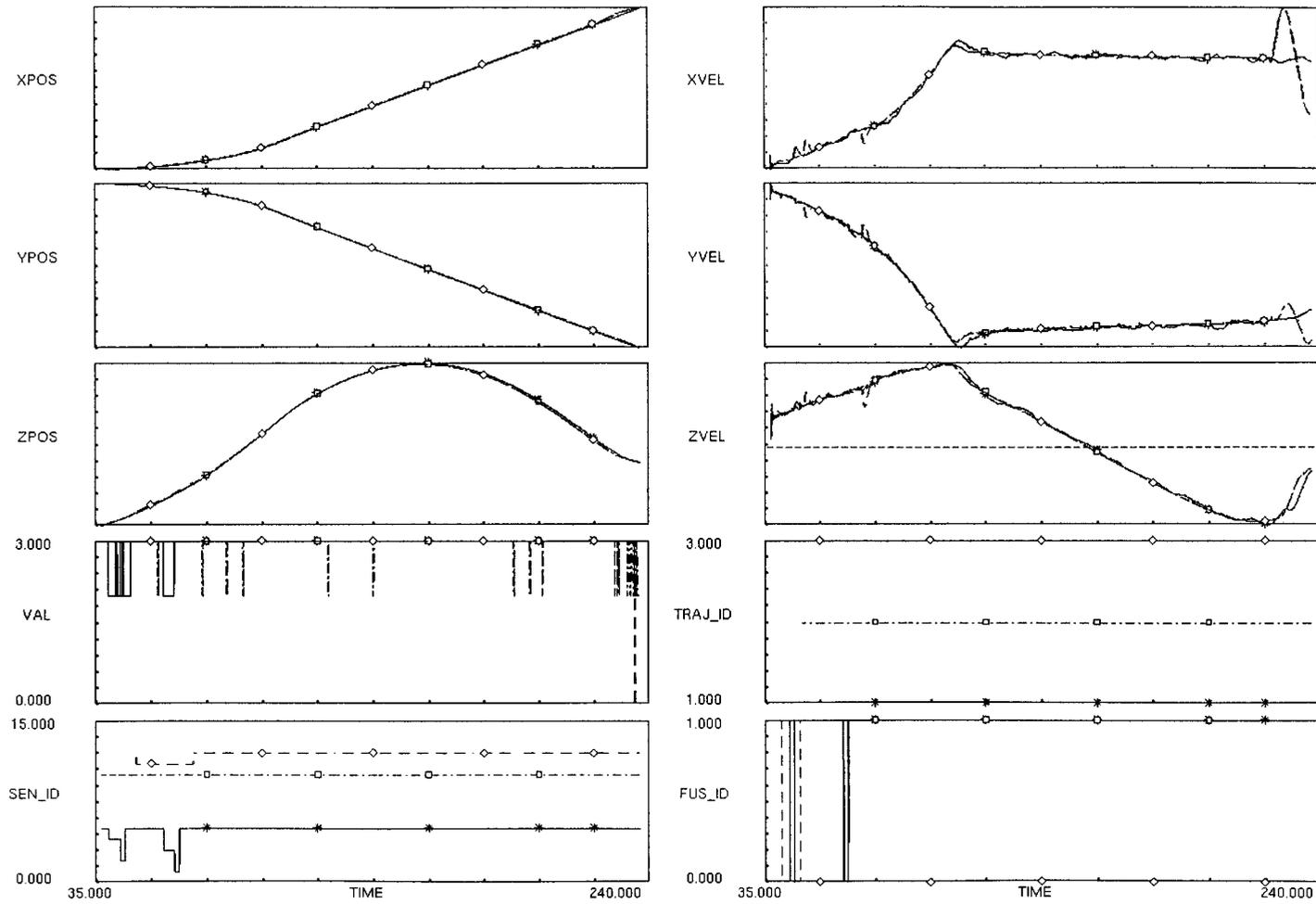


Figure 2. Comparison of TRAJ1, TRAJ2, and TRAJ3. Legend: —*— TRAJ1 - - - - □ - - - - TRAJ2 - - ◇ - - TRAJ3.

Fusion module 2

Code for the RADAR 1 and RADAR 2 sensor fusion.

Fusion module 3

Code for the GPS and INS sensor fusion.

4.2 Performance evaluation

The fusion philosophy (figure 1) is tested with one set of post flight data (where data is not available, some proxy data are used to carry out the validation). Figure 2 shows the comparison of TRAJ1, TRAJ2, and TRAJ3. The 'VAL' corresponds to validity bit/health of the sensor. TRAJ-ID stands for the trajectory identity signifying which module the trajectory belongs. SEN-ID gives the sensor identity during the trajectory of the target. The FUS-ID tells whether the trajectory is 'filtered only' or fused (after filtering). The possibilities given in tables 1–3 are exercised in generating the fused trajectories.

5. Concluding remarks

The *U-D* filtering and sensor fusion filter algorithms have been described and validated. A hierarchical fusion philosophy has been presented for a typical flight test range and tested for one set of post flight tracking data. Although the results are encouraging, further up-gradation of the process is under progress.

The authors are grateful to Mr Narang, Interim Test Range, Chandipur and Air Cmdre. P Banerjee for technical discussions and support during the course of this work.

References

- Bierman G J 1977 *Factorization methods for discrete sequential estimation* (New York: Academic Press)
- Girija G, Raol J R, Appavu Raj R, Banerjee P 1999 Intelligent multi sensor data fusion. Presented at the *National Seminar on Intelligent and Autonomous Systems*, Computer Society of India, Mumbai
- Hall D L 1992 *Mathematical techniques in multisensor data fusion* (Boston: Artech House)
- Saha R K 1996 Effect of common process noise on two-track fusion. *J. Guidance, Control Dyn.* 19: 829–835