Fusion Architectures for 3D target tracking using Radar and IRST Data

VPS Naidu and Girija G.
Multi Sensor Data Fusion Lab
Flight Mechanics and Control Division
National Aerospace Laboratories
Bangalore-17, India
E-mail: vpsnaidu@gmail.com

Abstract-**Six different architectures are presented to fuse IRST and radar data to track the target in 3D Cartesian coordinates, with the measurements available in polar coordinates. Performance of these architectures is evaluated with simulated data. Detailed mathematical expressions are provided which could be useful for algorithm implementation. From this study, it is concluded that SVF (state vector fusion) architecture provides state estimates with less uncertainty. However, the choice of a particular architecture for fusion of disparate sensor data is trade off between accuracy and computational complexity.**

I. INTRODUCTION

Modern military aircraft are equipped with disparate sensors to aid the pilot to perform his designated tasks in various missions. Radars and IRST are two of the sensors that are generally used in the cockpit for providing target information. In general, the sensor measurements are not perfect and are corrupted with noise. Moreover, a single sensor may not provide all the information required about the location and attitude of the target in target tracking applications. Hence, filters and multiple sensors are used to enhance the target tracking capabilities [1]. Radars provide measurements of azimuth, elevation and range of a target. It can measure range with good resolution, but the angular measurements are not so accurate. The uncertainty associated with radar might be represented as a volume whose dimensions are relatively large perpendicular to the measured line of sight and small along the line of sight. On the other hand, an infrared search and track sensor (IRST) can measure azimuth and elevation of a target with good resolution. The uncertainty associated with IRST could be represented as a square whose dimensions are comparatively small perpendicular to the measured line of sight. By fusing the measurements from radar and IRST, the resultant uncertainty in the estimated position of the target would be smaller than the uncertainty of either of the measurements alone [1,2].

This paper deals with tracking of target in 3D Cartesian coordinates using the measurements from radar and IRST sensors in polar coordinates. Extended Kalman filter is used to estimate the state of the target using appropriate target motion and measurement models. In this paper the performance of six different fusion architectures [3-11] are evaluated for fusion of the IRST and radar data for target tracking. The performance of these algorithms is evaluated using several metrics like percentage fit error (PFE), root mean square error in position (RMSPE), root sum square error in position (RSSPE) and mean absolute state error (MAE). Computational complexity of the various architectures is also evaluated.

II. EXTENDED KALMAN FILTER

A general motion model used in discrete extended Kalman filter for target tracking is [1, 3]:

$$X(k) = FX(k-1) + Gw(k-1) \quad (1)$$

$$z(k) = h(X(k)) + v(k) \quad (2)$$

where $X(k)$ is the state vector, F is the state transition matrix and G is the process noise gain matrix. The process noise $w(k)$ and the measurement noise $v(k)$ are assumed to be zero-mean, mutually independent, white, Gaussian with covariance Q and R respectively. $z(k)$ is the measurement vector at time k and $h(X(k))$ is a nonlinear function of the states computed at time k.

Linear Kalman filter could be used for target tracking if both the states and the measurements are in Cartesian coordinate system. Radar and IRST provide the measurements in a spherical coordinate system and the state vector is to be estimated in Cartesian coordinate system. Eq (2) is nonlinear and needs to be linearized to fit into the Kalman filter framework entailing the use of extended Kalman filter (EKF).

A. State Prediction

The state and state covariance matrix at time $k-1$ are predicted to time k as follows:

$$\begin{aligned} \tilde{X}(k \mid k-1) &= F\hat{X}(k-1 \mid k-1) \\ \tilde{P}(k \mid k-1) &= F\hat{P}(k-1 \mid k-1)F^T + GQG^T \end{aligned} \quad (3)$$

where $\hat{X}$ is the estimated state vector, $\hat{P}$ is the estimated state covariance matrix, $\tilde{X}$ is the predicted state and $\tilde{P}$ is the predicted state covariance matrix.

B. Measurement Update

Innovation: $e = z(k) - \tilde{z}(k \mid k-1)$ (4)

Innovation covariance: $S = H\tilde{P}(k \mid k-1)H^T + R$ (5)

where $\tilde{z}(k \mid k-1)$ is the predicted measurement and H is the linearized measurement matrix. The measurement update part consists of the following equations.

Filter gain: $K = \tilde{P}(k \mid k-1)H^T S^{-1}$ (6)

Updated state: $\hat{X}(k \mid k) = \tilde{X}(k \mid k-1) + Ke$ (7)

Updated state covariance: $\hat{P}(k \mid k) = [I - KH]\tilde{P}(k \mid k-1)$ (8)

C. Predicted Measurement and Linearised Measurement Matrix

Finite difference method is used to compute the linearized measurement matrix. Consider the state vector consisting of position, velocity and acceleration components in x-, y- and z-direction as $\begin{bmatrix} x & \dot{x} & \ddot{x} & y & \dot{y} & \ddot{y} & z & \dot{z} & \ddot{z} \end{bmatrix}$ (9)

The predicted state is given by:

$$\begin{bmatrix} \tilde{x} & \dot{\tilde{x}} & \ddot{\tilde{x}} & \tilde{y} & \dot{\tilde{y}} & \ddot{\tilde{y}} & \tilde{z} & \dot{\tilde{z}} & \ddot{\tilde{z}} \end{bmatrix} = \tilde{X}(k \mid k-1) \quad (12)$$

The predicted measurement when the measurement vector consists of only azimuth and elevation is:

$$\tilde{z}(k \mid k-1) = h\left[\tilde{X}(k \mid k-1)\right] = \begin{bmatrix} \tilde{\theta} & \tilde{\varphi} \end{bmatrix}^T \quad (13)$$

The predicted measurement when the measurement vector consists of azimuth, elevation and range is:

$$\tilde{z}(k \mid k-1) = h\left[\tilde{X}(k \mid k-1)\right] = \begin{bmatrix} \tilde{\theta} & \tilde{\varphi} & \tilde{r} \end{bmatrix}^T \quad (14)$$

Components in the predicted measurement are computed from the predicted state vector given in eq.(12).

$$\tilde{\theta} = \tan^{-1}\left(\frac{\tilde{y}}{\tilde{x}}\right) \qquad \tilde{\varphi} = \tan^{-1}\left(\frac{\tilde{z}}{\sqrt{\tilde{x}^2 + \tilde{y}^2}}\right) \quad (15)$$

$$\tilde{r} = \sqrt{\tilde{x}^2 + \tilde{y}^2 + \tilde{z}^2}$$

Finite difference Method [4]: Calculation of linearized measurement matrix can be accomplished by the finite difference method as given below:

$$H(k) = H_{ij} = \left.\frac{\partial h_i}{\partial x_j}\right|_{x=\tilde{X}(k|k-1)} = \frac{h_i(x_j + \Delta x_j) - h_i(x_j)}{\Delta x_j} \quad (16)$$

where $i = 1, 2, \ldots,$ length of the measurement vector

j=1,2,...,length of the state vector

Δx_j =perturbation step size

For small perturbation Δx in each of the unknown variables, the perturbed value $h_i(x_j + \Delta x_j)$ is computed. The corresponding elements of H_{ij} are given by the finite difference of the function h (eq. 2) to changes in that state. In general, a perturbation step size of 10^{-7} is considered to be adequate.

III. FUSION OF IRST AND RADAR DATA

In this section six different architectures are presented to fuse IRST and radar data to track the target in 3D Cartesian coordinates, where the measurements come from radar and IRST are in polar coordinates. $z_i(k) = \begin{bmatrix} \theta_i & \varphi_i \end{bmatrix}^T$ and $z_r(k) = \begin{bmatrix} \theta_r & \varphi_r & r_r \end{bmatrix}^T$ denote the measurements from IRST and radar respectively. Noise covariance matrix of IRST is: $R_i = diag\left[{}_i\sigma_\theta^2 \quad {}_i\sigma_\varphi^2\right]$

Noise covariance matrix of radar is:

$$R_r = diag\left[{}_r\sigma_\theta^2 \quad {}_r\sigma_\varphi^2 \quad {}_r\sigma_r^2\right]$$

A. Selective Measurements (SM)

In this architecture (see Fig-1), the measurement vector consists of selective measurements from radar and IRST. The measurement vector consists of azimuth and elevation measurements taken from IRST and range measurement taken from radar. Similarly, the measurement covariance matrix is formed. It is a simple EKF filter and the equations required to implement the tracking algorithm are as follows:

State Prediction:

$$\begin{aligned} \tilde{X}(k \mid k-1) &= F\hat{X}(k-1 \mid k-1) \\ \tilde{P}(k \mid k-1) &= F\hat{P}(k-1 \mid k-1)F^T + GQG^T \end{aligned} \quad (17)$$

Fusion:

$$\begin{aligned} z(k) &= \begin{bmatrix} \theta_i & \varphi_i & r_r \end{bmatrix}^T \\ R &= diag\left[{}_i\sigma_\theta^2 \quad {}_i\sigma_\varphi^2 \quad {}_r\sigma_r^2\right] \end{aligned} \quad (18)$$

Measurement updation:

$$\begin{aligned} H &= h(\tilde{X}(k \mid k-1)) \\ \tilde{z}(k \mid k-1) &= H\tilde{X}(k \mid k-1) \\ e &= z(k) - \tilde{z}(k \mid k-1) \\ S &= H\tilde{P}(k \mid k-1)H^T + R \\ K &= \tilde{P}(k \mid k-1)H^T S^{-1} \\ \hat{X}(k \mid k) &= \tilde{X}(k \mid k-1) + Ke \\ \hat{P}(k \mid k) &= [I - KH]\tilde{P}(k \mid k-1) \end{aligned} \quad (19)$$

B. Measurement Fusion (MF) [5,6]

In this architecture (see Fig-2), the measurement vector consists of fused azimuth, fused elevation and range taken from radar. Similarly, the measurement covariances for azimuth and elevation from IRST and radar are fused and measurement noise covariance for range is taken from radar. The fused azimuth is obtained by fusing the azimuths coming from IRST and radar. Similarly, the fused elevation is obtained by fusing the elevations coming from IRST and radar. Instead of fusing the measurement in the use of EKF, the measurements from IRST and radar are merged into an augmented measurement vector and measurement noise variances from both sensors also concatenated to produce the same results [5].

State Prediction: (eq. 17)

Fusion:

$$
\begin{aligned}
&z_d(k) = \begin{bmatrix}\theta_r & \varphi_r\end{bmatrix}^T \\
&R_d = diag\begin{bmatrix}{}_r\sigma_\theta^2 & {}_r\sigma_\varphi^2\end{bmatrix} \\
&z_f = z_i(k) + z_i(k)\left[R_i + R_d\right]^{-1}\left(z_d(k) - z_i(k)\right) \\
&R_f = R_i - R_i\left[R_i + R_d\right]^{-1}R_i \\
&z(k) = \begin{bmatrix}z_f(1) & z_f(2) & r_r\end{bmatrix}^T \\
&R = diag\begin{bmatrix}R_f(1,1) & R_f(2,2) & {}_r\sigma_r^2\end{bmatrix}
\end{aligned} \tag{20}
$$

Measurement updation: (eq.19)

C. State Vector Fusion (SVF) [5-9]

In this architecture (see Fig-3), tracks are formed with IRST and radar measurements separately and the resultant state vectors (tracks) are fused to get final target state estimations. Similarly, the state error covariances of the individual tracks are fused to get the final state error covariance matrix.

State Prediction:

$$
\begin{aligned}
&\tilde{X}_i(k \mid k-1) = F\hat{X}_i(k-1 \mid k-1) \\
&\tilde{P}_i(k \mid k-1) = F\hat{P}_i(k-1 \mid k-1)F^T + GQG^T
\end{aligned} \tag{21a}
$$

$$
\begin{aligned}
&\tilde{X}_r(k \mid k-1) = F\hat{X}_r(k-1 \mid k-1) \\
&\tilde{P}_r(k \mid k-1) = F\hat{P}_r(k-1 \mid k-1)F^T + GQG^T
\end{aligned} \tag{21b}
$$

Measurement updation:

$$
\begin{aligned}
&H_i = h(\tilde{X}_i(k \mid k-1)) \\
&\tilde{z}_i(k \mid k-1) = H_i\tilde{X}_i(k \mid k-1) \\
&e_i = z_i - \tilde{z}_i(k \mid k-1) \\
&S_i = H_i\tilde{P}_i(k \mid k-1)H_i^T + R_i \\
&K_i = \tilde{P}_i(k \mid k-1)H_i^T S_i^{-1} \\
&\hat{X}_i(k \mid k) = \tilde{X}_i(k \mid k-1) + K_i e_i \\
&\hat{P}_i(k \mid k) = \left[I - K_i H_i\right]\tilde{P}_i(k \mid k-1)
\end{aligned} \tag{22a}
$$

$$
\begin{aligned}
&H_r = h(\tilde{X}_r(k \mid k-1)) \\
&\tilde{z}_r(k \mid k-1) = H_r\tilde{X}_r(k \mid k-1) \\
&e_r = z_r - \tilde{z}_r(k \mid k-1) \\
&S_r = H_r\tilde{P}_r(k \mid k-1)H_r^T + R_r \\
&K_r = \tilde{P}_r(k \mid k-1)H_r^T S_r^{-1} \\
&\hat{X}_r(k \mid k) = \tilde{X}_r(k \mid k-1) + K_r e_r \\
&\hat{P}_r(k \mid k) = \left[I - K_r H_r\right]\tilde{P}_r(k \mid k-1)
\end{aligned} \tag{22b}
$$

Fusion:

$$
\begin{aligned}
&\hat{X}_f(k \mid k) = \hat{X}_i(k \mid k) \\
&\quad + \hat{P}_i(k \mid k)\left[\hat{P}_i(k \mid k) + \hat{P}_r(k \mid k)\right]^{-1}\left(\hat{X}_r(k \mid k) - \hat{X}_i(k \mid k)\right) \\
&\hat{P}_f(k \mid k) = \hat{P}_i(k \mid k) \\
&\qquad - \hat{P}_i(k \mid k)\left(\hat{P}_i(k \mid k) + \hat{P}_r(k \mid k)\right)^{-1}\hat{P}_i^T(k \mid k)
\end{aligned} \tag{23}
$$

D. Feedback SVF [5,10]

In this architecture (see Fig-4), the fused state vector and state error covariance matrix are fedback to a single state predictor and the out put of this is fed to two measurement updation. IRST measurements are used at one of the measurement updation to estimate the target states and radar measurements are used at the other measurement updation to estimate the target states. Finally the estimates are fused and then fedback to the prediction.

State Prediction:

$$
\begin{aligned}
&\tilde{X}(k \mid k-1) = F\hat{X}_f(k-1 \mid k-1) \\
&\tilde{P}(k \mid k-1) = F\hat{P}_f(k-1 \mid k-1)F^T + GQG^T
\end{aligned} \tag{24}
$$

Measurement updation: (eq. 22)

Fusion:

$$
\begin{aligned}
&P_{ir}(k \mid k) = P_{ri}^T(k \mid k) = \left[I - K_i(k)H_i(k)\right] \\
&\qquad \tilde{P}(k \mid k-1)\left[I - K_r(k)H_r(k)\right]^T
\end{aligned} \tag{25}
$$

$$
\begin{aligned}
&\hat{X}_f(k) = \hat{X}_i(k \mid k) + \left[\hat{P}_i(k \mid k) - \hat{P}_{ir}(k \mid k)\right] \\
&\left[\hat{P}_i(k \mid k) + \hat{P}_r(k \mid k) - \hat{P}_{ir}(k \mid k) - \hat{P}_{ri}(k \mid k)\right]^{-1}\left(\hat{X}_r(k \mid k) - \hat{X}_i(k \mid k)\right) \\
&\hat{P}_f(k \mid k) = \hat{P}_i(k \mid k) - \left[\hat{P}_i(k \mid k) - \hat{P}_{ir}(k \mid k)\right] \\
&\left[\hat{P}_i(k \mid k) + \hat{P}_r(k \mid k) - \hat{P}_{ir}(k \mid k) - \hat{P}_{ri}(k \mid k)\right]^{-1}\left[\hat{P}_i(k \mid k) - \hat{P}_{ir}(k \mid k)\right]
\end{aligned} \tag{26}
$$

E. Predicted SVF (PSVF) [5]

In this architecture (see Fig-5), the predicted state vectors from IRST and radar are fused. Similarly, the predicted state error covarainces are also fused. The fused estimates are fed to two measurement updation. IRST measurements are used in one of the measurement updation to estimate the target states and radar measurements are used in the other measurement updation to estimate the target states. These estimates are fedback to the respective prediction stage and also fused to get the final target estimates.

State Prediction: (eq. 21)

$$
\tilde{P}_{ir}(k \mid k-1) = \tilde{P}_{ri}^T(k \mid k-1) = F\hat{P}_{ir}(k-1 \mid k-1)F^T + GQG^T \tag{28}
$$

Fusion: (at predicted stage)

$$
\begin{aligned}
&\tilde{X}(k \mid k-1) = \tilde{X}_i(k \mid k-1) + \left[\tilde{P}_i(k \mid k-1) - \tilde{P}_{ir}(k \mid k-1)\right] \\
&\left[\tilde{P}_i(k \mid k-1) + \tilde{P}_r(k \mid k-1) - \tilde{P}_{ir}(k \mid k-1) - \tilde{P}_{ri}(k \mid k-1)\right]^{-1} \\
&\qquad \left(\tilde{X}_r(k \mid k-1) - \tilde{X}_i(k \mid k-1)\right) \\
&\tilde{P}(k \mid k-1) = \tilde{P}_i(k \mid k-1) - \left[\tilde{P}_i(k \mid k-1) - \tilde{P}_{ir}(k \mid k-1)\right] \\
&\left[\tilde{P}_i(k \mid k-1) + \tilde{P}_r(k \mid k-1) - \tilde{P}_{ir}(k \mid k-1) - \tilde{P}_{ri}(k \mid k-1)\right]^{-1} \\
&\left[\tilde{P}_i(k \mid k-1) - \tilde{P}_{ir}(k \mid k-1)\right]
\end{aligned} \tag{29}
$$

Measurement updation: (eq. 22)

$$
\begin{aligned}
&\hat{P}_{ir}(k \mid k) = \hat{P}_{ri}^T(k \mid k) = \left[I - K_i(k)H_i(k)\right] \\
&\qquad \tilde{P}(k \mid k-1)\left[I - K_r(k)H_r(k)\right]^T
\end{aligned} \tag{31}
$$

Fusion: (at estimated stage)

$$X(k) = \hat{X}_i(k|k) + \left[\hat{P}_i(k|k) - \hat{P}_{ir}(k|k)\right]$$
$$\left[\hat{P}_i(k|k) + \hat{P}_r(k|k) - \hat{P}_{ir}(k|k) - \hat{P}_{ri}(k|k)\right]$$
$$\left(\hat{X}_r(k|k) - \hat{X}_i(k|k)\right)$$
$$P(k) = \hat{P}_i(k|k) - \left[\hat{P}_i(k|k) - \hat{P}_{ir}(k|k)\right]$$
$$\left[\hat{P}_i(k|k) + \hat{P}_r(k|k) - \hat{P}_{ir}(k|k) - \hat{P}_{ri}(k|k)\right]^{-1}$$
$$\left[\hat{P}_i(k|k) - \hat{P}_{ir}(k|k)\right]$$

F. Decentralized Kalman Filter [11-13]

In this algorithm, the states obtained from local Kalman filters (LKF) are fed to the global Kalman filter (GKF) for final target estimates as shown in Fig-6. One of the local KF utilizes IRST measurements and another local KF utilizes radar measurements. The LKFs transmit only the state error information ($\hat{X}_i(k)$ & $\hat{X}_r(k)$) and covariance error information ($\hat{P}_i(k)$ & $\hat{P}_r(k)$) to the GKF. The GKF have state prediction and estimate correction instead of measurement updation. At each local EKF the following quantities has to be computed and then passed to the global EKF. The state and covariance error information are utilized at estimate correction stage to obtain final target estimates.

At LKF: (using IRST measurements)

State prediction: (eq. 19a)

Measurement updation: (eq. 22a)

$$\hat{X}_i(k) = \hat{P}_i^{-1}(k|k)\hat{X}_i(k|k) - \tilde{P}_i^{-1}(k|k-1)\tilde{X}_i(k|k-1)$$
$$\hat{P}_i(k) = P_i^{-1}(k|k) - P_i^{-1}(k|k-1) \quad (34)$$

At LKF: (using radar measurements)

State prediction: (eq. 19b)

Measurement updation: (eq. 22a)

$$\hat{X}_r(k) = \hat{P}_r^{-1}(k|k)\hat{X}_r(k|k) - \tilde{P}_r^{-1}(k|k-1)\tilde{X}_r(k|k-1)$$
$$\hat{P}_r(k) = P_r^{-1}(k|k) - P_r^{-1}(k|k-1) \quad (35)$$

Fusion: **(at GKF)**

State prediction:

$$\tilde{X}(k|k-1) = F\hat{X}(k-1|k-1)$$
$$\tilde{P}(k|k-1) = F\hat{P}(k-1|k-1)F^T + GQG^T \quad (36)$$

Estimate correction:

$$\hat{P}^{-1}(k|k) = \tilde{P}^{-1}(k|k-1) + \hat{P}_i(k) + \hat{P}_r(k) \quad (37)$$
$$\hat{X}(k|k) = \hat{P}(k|k)\left[\tilde{P}^{-1}(k|k-1)\tilde{X}(k|k-1) + \hat{X}_i(k) + \hat{X}_r(k)\right]$$

IV. NUMERICAL SIMULATION AND DISCUSSION

The 3DOF kinematic model, with position, velocity and acceleration components in each of the three Cartesian coordinates x, y and z has the following transition and process noise gain matrices:

$$F = diag\left[\Phi \quad \Phi \quad \Phi\right] \qquad G = diag\left[\zeta \quad \zeta \quad \zeta\right] \quad (38)$$

where $\Phi = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}$ $\zeta = \begin{bmatrix} T^3/6 \\ T^2/2 \\ T \end{bmatrix}$

where T is a sampling interval, F is the state transition matrix and G is the process noise gain matrix

Simulation utilizes the following parameters:

Sampling interval: 0.1sec.

Process noise variance: 1^2

Measurement noise variance: shown in Table I

Duration of simulation: 50sec.

Initial state vector is:

$[x \quad \dot{x} \quad \ddot{x} \quad y \quad \dot{y} \quad \ddot{y} \quad z \quad \dot{z} \quad \ddot{z}] =$

$[10000 \quad -200 \quad 0.5 \quad -1000 \quad -100 \quad -0.3 \quad 1000 \quad 1 \quad 0.01]$

The simulated noisy measurements in polar coordinates are shown in Fig-7. The initial state vector is chosen as:

$$\hat{X}_0 = 0.9X_t \quad (39)$$

$\hat{X}_0$: initial estimated state vector at scan number one

X_t : true state vector at scan number one

The expression for the initial state error covariance matrix is given by: $\hat{P}_0 = \text{diag}[(X_t - \hat{X}_0)^2]$ (40)

The filter performance is checked by computing [7]:

i.The percentage fit error (PFE) in x, y & z positions:

$$PFEx = 100 * \frac{norm(x - \hat{x})}{norm(x)}$$, similarly for y and z pos. (41)

ii.Root mean square error in position:

$$RMSPE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\frac{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (z_i - \hat{z}_i)^2}{3}} \quad (42)$$

iii.Root sum square error in position:

$$PSSPE = \sqrt{(x - \hat{x})^2 + (y - \hat{y})^2 + (z - \hat{z})^2} \quad (43)$$

iv.Absolute error in (AE) x, y & z positions:

$AE_x(i) = |x(i) - \hat{x}(i)| \quad i = 1,2,...,N$, similarly for y and z positions (44)

v.Mean absolute error in x, y & z positions:

$$MAE_x = \frac{1}{N}\sum_{i=1}^{N}|x(i) - \hat{x}(i)|$$, similarly for y and z pos. (45)

Performance of six fusion architectures is evaluated using fifty Monte Carlo simulations. The percentage fit error (PFE) in x-, y- and z-positions are shown in Table II and mean absolute error in x-, y- and z-position and their derivatives are shown in Table III. Root mean square error in position, velocity and acceleration are shown in Table IV. The values shown in bold indicate the best results. The root sum position error and root sum velocity error are shown in Fig-8a and Fig-8b respectively. Absolute error in x-, y- and z-positions are shown in Fig-9. From tables and Figs-8to9, it is observed that DKF performance is poor and SVF performance is better in velocity and acceleration estimates. Overall, SM performance is very good. Root sum variance and mean root sum variance related position, velocity & acceleration are shown in Fig-10

and Table V respectively. It is observed that, SVF shows the lowest uncertainty followed by DKF and other architectures show high uncertainty in the state estimation.

V. CONCLUSION

Six different architectures are presented to fuse IRST and radar data to track the target in 3D Cartesian coordinates, where the measurements are in polar coordinates. Their performance is evaluated with numerical simulation. Detailed mathematical expressions are given which could be useful for implementation. From the results it is concluded that SVF architecture provides state estimates with less uncertainty. The fusion architectures presented in this paper are easy to implement and could used in real time. However, the choice of a particular architecture for fusion of disparate sensor data is trade off between accuracy and computational complexity.

REFERENCES

[1] Samuel Blackman and Robert Popoli, "Design and Analysis of Modern Tracking Systems", Artech House, London, 1999.

[2] David L. Hall and Sonya A. H. McMullen, "Mathematical Techniques in Multisensor Data Fusion", Second edition, Artech House, London, 2004.

[3] Yaakov Bar-Shalam and X. Li, "Estimation and Tracking: Principles, Techniques, and Softwares", Artech House, London, 1993.

[4] Girija G. and Christoph Zorn, "Flight Path Reconstruction for Sensor Failure Detection and Health Monitoring", NAL PD FC 0418, 2004.

[5] J.B. Gao and C.J. Harris, "Some Remarks on Kalman Filters for the Multisensor Fusion", Information Fusion, 3, pp.191-201, 2002.

[6] Zhiqiang Hou and Chonghao Han, "A Target Tracking System based on Radar and Image Fusion", International Conference on Information Fusion, Australia, pp. 1426-1432, 2003.

[7] VPS Naidu, G. Girija and JR Raol, "Data Association and Fusion Algorithms for Tracking in presence of Measurement Loss", Journal of the Institution of Engineers (I), Vol. 86, pp.17-28, May 2005.

[8] R. K. Saha, "Effect of Common Process Noise on Two-Sensor Track Fusion", J. of Gui and Dyn., Vol.19, No.4, pp.829-835, July, 1996.

[9] J. Roecker and C. McGillem, "Comparison of Two-sensor Tracking Methods based on State Vector Fusion and Measurement Fusion", IEEE Trans. on Aero. And Elec. Sys., 24(4), pp.447-449, 1988.

[10] C. Blanc, L.Trassoudaine, Y. Le Guilloux and R. Moreira, "Track to Track Fusion Method Applied to Road Obstacle Detection", International Conference on Information Fusion, Stockholm, Sweden, pp. 775-782, 2004.

[11] B.S.Y. Rao, H.F. Durrant-Whyte and J.A. Sheen, "A Fully Decentralized Multi-Sensor System for racking and Surveillance", The International Journal of Robotics Research, 12(1), pp.20-44, 1993.

[12] Hamid R. Hashemipour, Sumit Roy and Alan J. Laub, "Decentralized Structures for Parallel Kalman Filtering", IEEE Trans. on Automatic Control, 33(1), pp88-94, 1988.

[14] N. Strobel, S. Spors and R. Rabenstein, "Joint Audio-Video Object Localization and Tracking", IEEE Signal Processing Magazine, pp.22-31, 2001.

TABLE I
MEASUREMENT NOISE VARIANCE

Sensor	Azimuth(rad)	Elevation(rad)	Range (m)
IRST	${}_i\sigma_\theta^2 = 10^{-5}$	${}_i\sigma_\varphi^2 = 10^{-5}$	-------
Radar	${}_r\sigma_\theta^2 = 10^{-2}$	${}_r\sigma_\varphi^2 = 10^{-2}$	${}_r\sigma_r^2 = 100$

TABLE II
PERCENTAGE FIT ERRORS IN X-, Y- AND Z-POSITION

architecture	*PFEx*	*PFEy*	*PFEz*
SM	**0.7850**	**0.1646**	0.7078
MF	**0.7850**	0.1648	0.7079
SVF	0.7855	0.1684	**0.7031**
FSVF	**0.7850**	**0.1646**	0.7078
PSVF	**0.7850**	0.1650	0.7080

TABLE III
MEAN ABSOLUTE ERRORS IN X-, Y- AND Z-POSITION AND THEIR DERIVATIVES

architecture	*MAEx*	*MAEy*	*MAEz*	*MAEẋ*	*MAEẏ*	*MAEż*	*MAEẍ*	*MAEÿ*	*MAEz̈*
SM	6.2808	4.9083	5.7457	3.3362	3.1445	3.1450	1.4132	1.2006	1.2951
MF	**6.2790**	4.9125	5.7482	3.3344	3.1471	3.1465	1.4129	1.2008	1.2955
SVF	6.4871	5.1671	**5.6798**	**3.3188**	**3.0972**	**2.9950**	**1.3651**	**1.1348**	**1.2297**
FSVF	6.2808	**4.9080**	5.7453	3.3362	3.1455	3.1449	1.4132	1.2006	1.2950
PSVF	6.2994	4.9327	5.7456	3.3615	3.1330	3.1446	1.4129	1.1799	1.2956
DKF	6.5392	5.1775	5.7732	3.3933	3.2296	3.1563	1.4196	1.2264	1.2972

TABLE IV
ROOT MEAN SQUARE ERRORS IN POSITION, VELOCITY AND ACCELERATION

architecture	RMSPE	RMSVE	RMSAE
SM	26.8065	4.2333	1.6684
MF	26.8072	4.2347	1.6686
SVF	26.8345	**4.1580**	**1.5979**
FSVF	**26.8064**	4.2331	1.6684
PSVF	26.8101	4.2282	1.6610
DKF	26.8519	4.3008	1.6833

TABLE V
MEAN ROOT SUM VARIANCE IN POSITION, VELOCITY AND ACCELERATION

architecture	MRSvarP	MRSvarV	MRSvarA
SM	13.0025	7.4623	3.1682
MF	12.9984	7.4609	3.1680
SVF	**12.3739**	**6.5193**	**2.4862**
FSVF	13.0026	7.4624	3.1682
PSVF	12.9532	7.3113	3.1667
DKF	12.7399	7.3659	3.1565

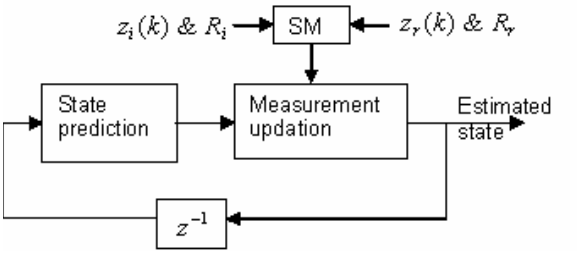

Fig-1. Information flow diagram of SF algorithm

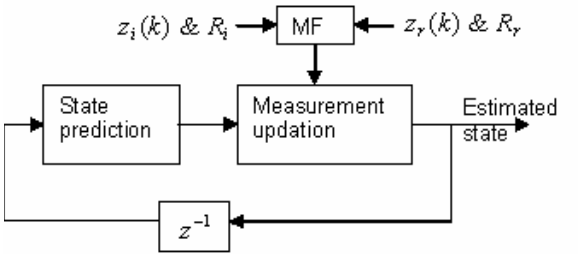

Fig-2. Information flow diagram of MF algorithm

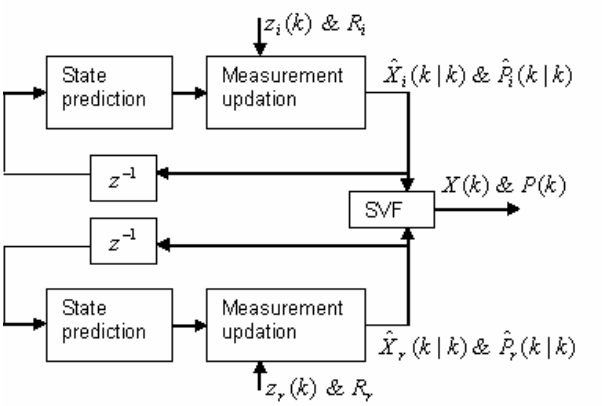

Fig-3. Information flow diagram of SVF algorithm

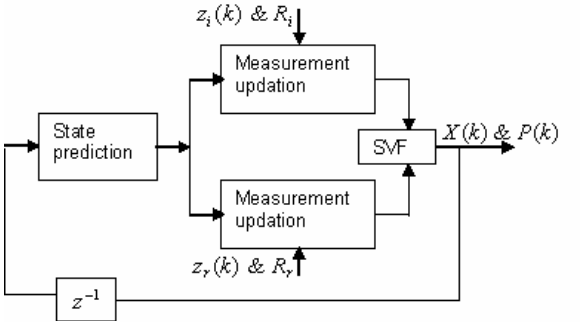

Fig-4. Information flow diagram of FSVF algorithm

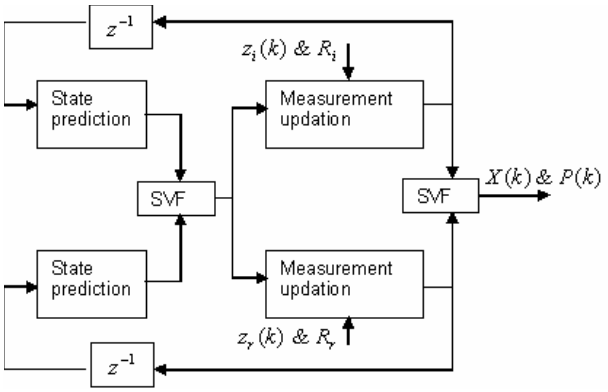

Fig-5. Information flow diagram of PSVF algorithm

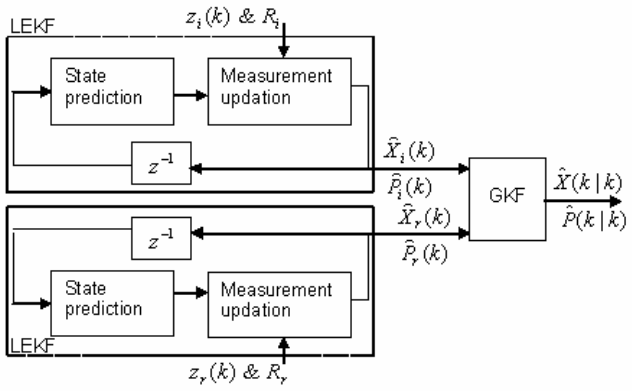

Fig-6. Information flow diagram of DF algorithm

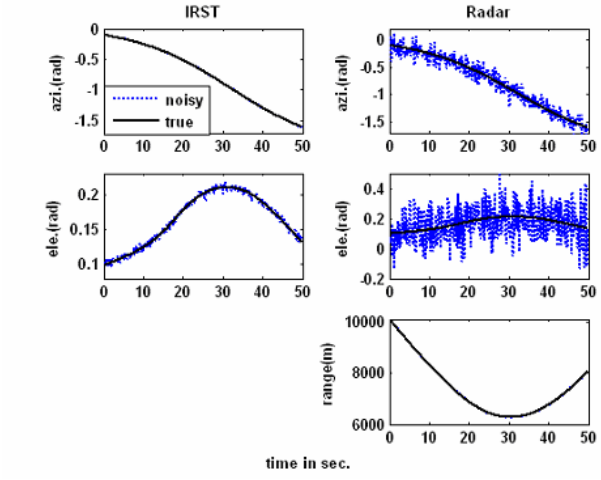

Fig-7. Noisy measurements in polar coordinates

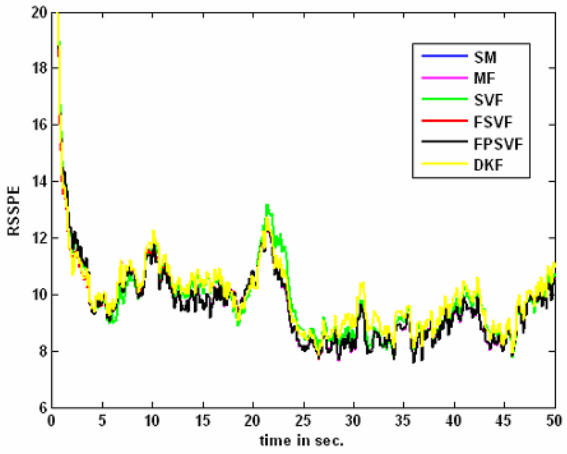

Fig-8a. Root sum square position error

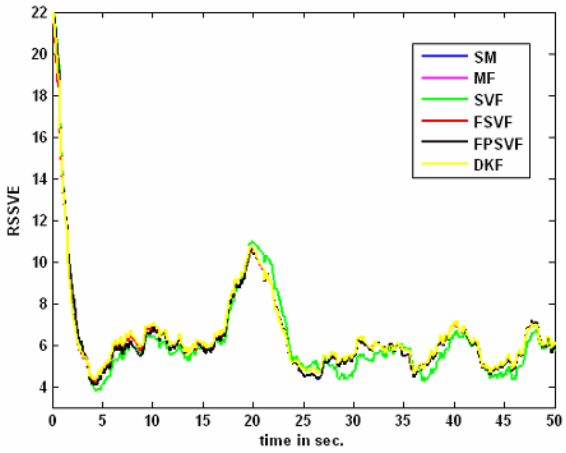

Fig-8b. Root sum square velocity error

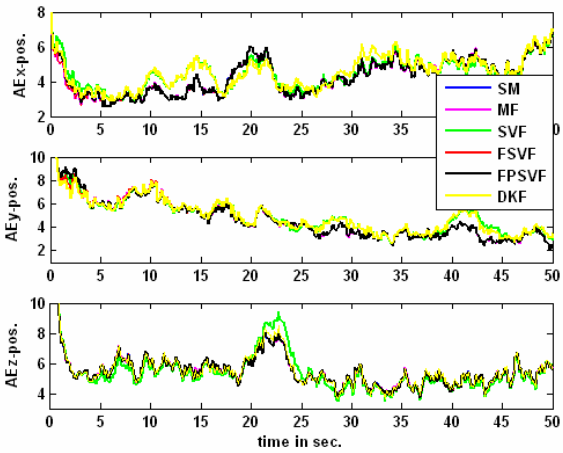

Fig-9. Absolute error in x-,y- and z-position

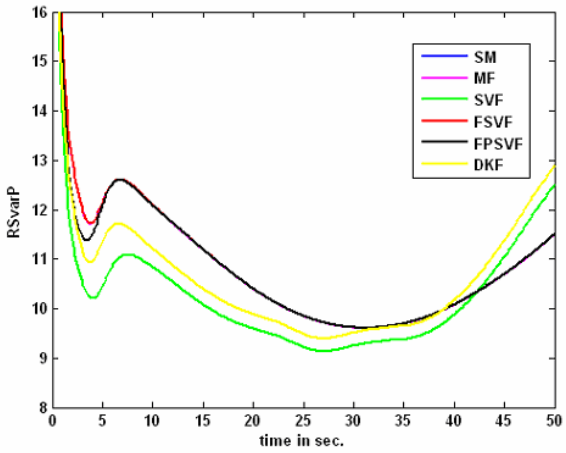

Fig-10a. Root sum variance in position

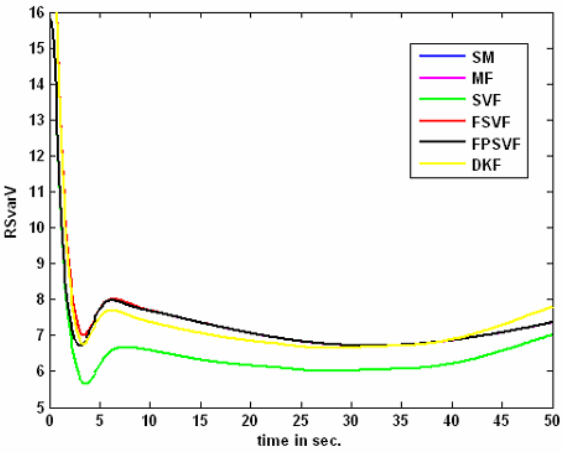

Fig-10b. Root sum variance in velocity

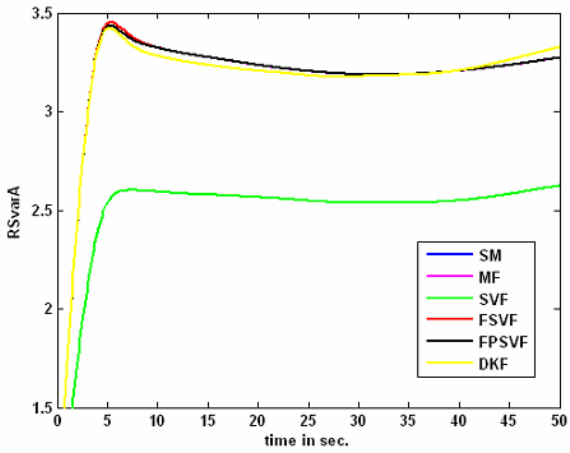

Fig-10c. Root sum variance in acceleration