## Multi Sensor Data Fusion Algorithms and Architectures: Literature Review

**Abstract**

An important distinction between theoretical and practical worlds of communication and control, are the statistical problems of: (i) prediction of random signals; (ii) separation of random signals from random noise; (iii) detection of signals of known form (pulses, sinusoids) in the presence of random noise. R.E. Kalman (1960) writes how sensors are not perfect and their measurements are corrupted with noise. Moreover, single sensor may not provide all the information about the target. Hence, filters and multiple sensors are used to enhance the target tracking capabilities.

For example, for 3D tracking of objects in Cartesian coordinates by aircrafts, Infra-Red Search and Tracking (IRST) and RADAR are used. VPS Naidu (2010) explains the shortcomings of both these sensors. Radar can measure azimuth, elevation and range of a target. It can measure range with good resolution, but the angular measurements are not so accurate. Despite this, radar provides sufficient information to track the target. 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. An infrared search and track sensor (IRST) can measure azimuth and elevation of a target with good resolution. It can provide the direction of a target but not its location because it does not measure the range. The uncertainty associated with IRST might 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 of the estimated position of the target would be smaller than the uncertainty of either of the measurements alone.

Multi Sensor Data Fusion can be done at four different processing levels as discussed by Jiang Dong *et al* (2009): (i) Signal level fusion. In signal-based fusion, signals from different sensors are combined to create a new signal with a better signal-to noise ratio than the original signals. (ii) Pixel level fusion. Pixel-based fusion is performed on a pixel-by-pixel basis. It generates a fused image in which information associated with each pixel is determined from a set of pixels in source images to improve the performance of image processing tasks such as segmentation. (iii) Feature level fusion. Feature-based fusion at feature level requires an extraction of objects recognized in the various data sources. It requires the extraction of salient features which are depending on their environment such as pixel intensities, edges or textures. These similar features from input images are fused. (iv) Decision level fusion consists of merging information at a higher level of abstraction, combines the results from multiple algorithms to yield a final fused decision. Input images are processed individually for information extraction. The obtained information is then combined applying decision rules to reinforce common interpretation. In this paper, signal level fusion architectures are discussed and reviewed.

The first step in multi sensor data fusion is making sure the different sensor data and prediction algorithm output are read at a coordinating frequency. Based on the physical attributes to be estimated from the sensors, a state and measurement model is designed. The state model is governed by mathematical formulae of physics. Weiner process acceleration model is mostly chosen as the state model for navigation projects to estimate position, velocity and acceleration in the different axes of motion.



Each signal value xk is evaluated and the next value for the signal is a linear combination of the linearized previous state value and a control signal uk along with process noise wk. zk denotes the measurement values which is mathematically the sum of xk (signal value) and vk (measurement noise). Both process noise and measurement noise are considered to be Gaussian. If a linear signal is fed to a Gaussian then the output is also Gaussian but if the Gaussian is fed with a non-linear function then the output leads to non-Gaussian distributions. RE Kalman(1960) first proposed the idea of the Kalman Filter which is a filter that works as a least square error optimizer by calculating mean and variance, and, for this to work, it is necessary that the system that you consider inside the filter is linear. In real life scenarios most of the signals we measure are non-linear in nature. Thus, comes the first challenge, which is to linearize the system. Based on the research up till now, this is tackled by non-linear Kalman filters such as: (i) Extended Kalman Filter (EKF); (ii) Unscented Kalman Filter (UKF); (iii) Cubature Kalman Filter (CKF).

**Keywords**

Kalman Filter; Extended Kalman Filter; Unscented Kalman Filter; Multi Sensor Data Fusion;

**Abbreviations**

EKF: Extended Kalman filter

UKF: Unscented Kalman Filter

CKF: Cubature Kalman Filter

GPS: Global Positioning System

IRST: Infra-Red Search and Track

**Steps of Multi Sensor Data Fusion**

1. Identify State Model according to the sensor measurement and data to be calculated
2. Kalman Filtering
   1. Linearization of State
   2. Based on previous reading and state model equations, Prediction of reading at t+1
   3. Get sensor Measurement reading and compare it with predicted data
   4. Update based on prediction or measured reading with respect to Kalman Gain calculated
3. Fuse sensor data in case of multiple sensors

**Comparison between Kalman filter algorithms for non-linear systems**

In order to make state estimation on nonlinear systems, or parameter estimation, using the Kalman filter, one of the possible approaches is to linearize the system under investigation around its current state and force the filter to use this linearized version of your system as a model. This is the **Extended Kalman Filter, or EKF**.

EKF linearizes the system by a powerful approximation tool of Taylor Series which helps us to get a linearized approximation of a non-linear function. EKF takes the first derivative of the non-linear function according to the Taylor series and draws a tangent around the mean of the Gaussian giving us an approximate linear function.

**Prediction**



**Updation**



Next we do the iterative process of predicting and updating common also to linear kalman filter. In Predict we just predict the new value () called predicted value based on the initial value and then predict the variance () in our prediction according to the various process noises present in the system.

In Update, we take in account the actual measurement coming from the device and we call this as measured value (). Here we calculate the difference between our predicted value and measured value, () and then decide which value to keep by calculating the Kalman Gain (). We then calculate the new value () and new uncertainty/error/variance () based on our decision made by Kalman Gain. These calculated values will finally be the predictions done by our Kalman Filter in iteration 1.

The output of the update step is again fed into the Predict State and the cycle goes on until the error/uncertainty between our predicted and actual values tends to converge to zero.

However, the EKF is not very stable and many times, when it does converge to the "right" solution, it is quite slowly. In order to improve the performance of this filter, instead of using linearization to predict the behavior of the system under investigation, the Unscented Transformation was suggested by authors, Julier and Uhlmann (2004). Hence, the Kalman Filter with the Unscented transformation is called **Unscented Kalman Filter, or UKF**.

EKF approximates the entire non-linear function using just one point from the Gaussian which is its mean which leads to inefficiency in performance time on the other hand, using all points from the Gaussian will result in computational problems. UKF solves this by proposing the idea of calculating weighted () sigma point matrix () using the mean of the Gaussian (), scaling factor (), dimensionality of the system () and the covariance matrix (). UKF take some points on source Gaussian and map them on target Gaussian according to their calculated weights after passing points through some non-linear function. The new mean and variance of transformed Gaussian is then calculated. This is called the Unscented Transform analogous to the predict step in the EKF.



*for i = 1 to n*

*for i = n+1 to 2n*

*for i = 1 to 2n*

The updated mean () and the variance () for the signal is calculated again;



For the update step, the sigma points are mapped to the measurement space by transform matrix () giving ().



The Jacobian matrix using the Taylor series need not be calculated in this step since the linearization step has been taken care of already using the Unscented transform. Next to calculate the error in prediction the cross correlation between sigma-points in state space and sigma points in the measurement space is calculated and finally initial estimates (and) are updated in an iterative process. calculated is the Kalman Gain which updates iteratively according to the variance observed in the predicted and the measured signals.



This filter has some advantages when compared to the EKF, because the Unscented transformation somehow describes the nonlinear system better than the linearization, hence this filter converges to the right solution more rapidly. However, as the EKF, this filter may become unstable and results may be biased.

Similar research proposes other renditions of the algorithm such as the **Cubature Kalman Filter (CKF)** first proposed by Ienkaran Arasaratnam and Simon Haykin in 2009. The filter is based on spherical-radial cubature rule to save time complexity in solving integrals.

**Kalman Filter**: It is a tool to predict values using a bunch of mathematical equations under the assumptions that our data is in the form of Gaussian Distribution and we apply linear equations to that Gaussian distribution.

**Extended Kalman Filter**: In real world, we have non-linear equations, because we may be predicting in one direction while our sensor is taking reading in some other direction, so it involves angles and sine cosine functions which are non-linear. So EKF takes helps of Taylor Series (and Jacobian Matrix further) to linearly approximate a non-linear function around the mean of the Gaussian and then predict the values. These approximations, however, can introduce large errors in the true posterior mean and covariance of the transformed (Gaussian) random variable, which may lead to sub-optimal performance and sometimes divergence of the ﬁlter.

**Unscented Kalman Filter**: The unscented transformation (UT) is a method for calculating the statistics of a random variable which undergoes a non-linear transformation. UT when extended to the iterative kalman filter algorithm captures the posterior mean and covariance accurately to the 3rd order (Taylor series expansion) for any non-linearity. Furthermore, the overall number of computations are the same order as the EKF.

**Comparison between fusion algorithms for multi sensor non-linear system**

VPS Naidu (2010) discusses and evaluates seven different fusion algorithm for two non-linear sensor measurements, RADAR and IRST sensor data fusion. The performance of these algorithms is presented in terms of percentage of fit error (PFE), root mean square error in position (RMSPE), root sum square error in position (RSSPE) and mean absolute state error (MAE).

Selective Measurement (SM) fusion architecture takes the selected attribute from each multi sensor data and feeds them to a simple EKF filter. For eg: in case of fusing IRST and RADAR data, azimuth and elevation measurements taken from IRST and range measurement taken from radar and fed to a non-linear KF.

In Measurement Fusion (MF) architecture the measurement vector and measurement covariance consist of fused data from the different sensors. Instead of fusing the measurement in the use of EKF, the measurements from the sensors are merged into an augmented measurement vector and measurement noise variances from sensors fed into a non-linear KF.

In State Vector Fusion (SVF) architecture, data is predicted using Kalman filter from all separate sensors giving the filter corrected data and covariance for each sensor. The data from each filtered sensor is then fused giving one fused data and single fused covariance.

In Feedback State Vector Fusion (FSVF) architecture the fused state vector and state error covariance matrix are fed back to a single state predictor and the output of this is fed to two measurement updation. Finally the estimates are fused and then fedback to the prediction.

In Predicted State Vector Fusion (PSVF) architecture the predicted state vectors from different sensors are fused. Similarly, the predicted state error covarainces are also fused. The fused estimates are feed to two measurement updation. These estimates are fedback to the respective prediction stage and also fused to get the final target estimates.

In Decentralized Kalman Filter (DKF) architecture the states obtained from local sensor level Kalman filters (LKF) are fed to the global Kalman filter (GKF) for final state estimates. The LKFs transmit only the state error information and covariance error information to the GKF. The GKF have state prediction and estimate correction instead of measurement updation. At each LKF the following quantities has to be computed and then passed to the GKF. The state and covariance error information are utilized at estimate correction stage to obtain final target estimates.

Among the fusion architectures, DKF performance is poor and SVF performance is better in velocity and acceleration estimates. Overall, SM performance is very good and SVF shows the lowest uncertainty followed by other architectures show high uncertainty in the state estimation.

**References**

[1] V. Naidu, “Fusion architectures for 3d target tracking using irst and radar measurements,” Journal Of Aerospace Sciences Technologies, vol. 62, no. 3, pp. 183–195, 2009.

[2] E.A.Wan, R.V.D. Merwe “The unscented Kalman filter for nonlinear estimation,”Adaptive Systems for Signal Processing, Communications, and Control Symposium, 2000.

[3] T.Bao, Z.Zhang, M.F.Sabahi, “An improved radar and infrared sensor tracking fusion algorithm based on IMM-UKF,” International Conference on Networking, Sensing and Control, 2019.

[4] A.C.Babu, R.K.Karri, M.S.Nisha, “Sensor Data Fusion Kalman Filter,” International conference on Design Innovations for 3Cs Compute Communicate Control, 2018.

[5] R.E.Kalman, “A New Approach to Linear Filtering and Prediction Problems,” ASME–Journal of Basic Engineering, vol.82, series-D, pp. 35–45, 1960.

[6] M.Mallick, K.C.Chang, S.Arulampalam, Y.Yan, “Heterogeneous Track-to-Track Fusion in 3D Using IRST Sensor and Air MTI Radar,” IEEE Transactions on Aerospace and Electronic Systems, 2019.

[7] D.Hong-de, D.Shao-wu, C.Yuan-cai, W.Guang-bin, “Performance Comparison of EKF/UKF/CKF for the Tracking of Ballistic Target,” TELKOMNIKA, vol.10, no. 3, pp. 1537-1542, 2012.

[8] K. Rajeswari, A. Ishwarya, K.K. Vaishnavi, S.J.Thiruvengadam, “Performance Analysis of Data Fusion Methods for Radar and IRST 3D Target Tracking,” IEEE WiSPNET, 2017.

[9] J. Dong, D. Zhuang, Y. Huang, J. Fu, “Advances in Multi-Sensor Data Fusion: Algorithms and Applications,” Sensors, pp. 7771-7784, 2009.