EVALUATION OF DERIVATIVE FREE KALMAN FILTER FOR NON-LINEAR STATE-PARAMETER ESTIMATION AND FUSION

S.K. Kashyap* and J.R. Raol**

Abstract

The estimation of the states-parameters of non-linear system is often carried out using Extended Kalman Filter (EKF). The EKF is only reliable for systems that are almost linear on the time scale of the updates (i.e. sampling interval). The limitation of EKF can be overcome by use of another class of recursive estimator named derivative free Kalman filter (DFKF) or more popularly known as Unscented Kalman filter, a method that propagates mean and covariance using non-linear transformation. In this paper two methods: i) factorized version of EKF (UD Extended Kalman Filter or UDEKF) and ii) DFKF are studied and evaluated using various sets of simulated data of the non-linear systems as well as one real data set. Sensitivity study of DFKF with respect to tuning parameters such as α, β, *and* κ *(used in creation of sigma points and their associated weights) is also carried out using one set of simulated data. DFKF as compared to EKF is more accurate, easier to implement and has same order of calculations. The concept of DFKF is extended to data fusion (DF) for similar sensors and algorithm is named DF-DFKF. Application of DFKF is demonstrated in parameter estimation problem.*

Key words*:Non-linear systems, Target tracking, Kalman filter, Extended Kalman filter, UD factorization, Derivative free transformation, Derivative free Kalman filter, Kinematic consistency, Parameter estimation, Data fusion*

Nomenclature

T	= sampling interval
$h(.)$	= non-linear sensor model
Z_m	= sensor measurement vector
W	= weight of sigma point
P	= state error covariance matrix
K	= filter gain
ρ	= air density
$C_{(.)}$	= aerodynamic force and moment coefficients
$C_{(.)_{(.)}}$	= aerodynamic force and moment derivatives
$f(.)$	= non-linear plant model
Z_t	= measurement vector with out noise
χ	= state vector of sigma point
n_a	= augmented state vector dim. of DFKF
S	= innovation covariance matrix
I_{yy}	= moment of inertia
$\bar{c}$	= mean aerodynamic chord

Introduction

Kalman Filter (KF) algorithm is used for recursive estimation of any linear system being observed by single or group of sensors with linear models, however, it is found that in many practical situations, system dynamics or/and sensor model have non-linear characteristics. Therefore, in order to use Kalman filter for such cases, the non-linear system or sensor model needs to be linearised with respect to predicted/estimated states at every instant of time and this results in Extended Kalman Filter (EKF) approach.

EKF has been extensively used for many non-linear applications and is widely accepted by estimation community for more than three decades. EKF algorithm provides only sub-optimal solution to a given non-linear estimation problem and it has been observed that algorithm has major two limitations such as:

* Scientist ** Head (Retd)
Flight Mechanics and Control Division, National Aerospace Laboratories, Kodihalli, Bangalore-560 017, India
Email : sudesh@css.nal.res.in
Manuscript received on 10 Apr 2007; Paper reviewed, revised and accepted on 06 Feb 2008

- The derivations of the Jacobian matrices (in case of linearization of sensor model), in most applications, are nontrivial and that can lead to implementation problem.

- Linearization can lead to highly unstable filter (even divergence) if the assumption that system is almost linear (local linearity) on the time scale of the updates (i.e. sampling interval) is violated. The filter divergence for highly non-linear system can occur only when the update interval is large and within that interval the system is treated as a linear one but in reality if it is non-linear.

In target tracking application, sensor often provides non-linear measurements in Polar frame i.e. range, bearing or azimuth and elevation, whereas the state estimation is performed in Cartesian frame. In such circumstances, state estimation can be performed in several ways. For e.g. in converted measurement Kalman filter (CMKF), the measurements are transformed from Polar frame to Cartesian frame and then these converted measurements are used for state estimation. The major drawback of this approach is that it introduces measurements inaccuracies due to transformation, especially when the cross range error is more. The other way is to use EKF but it gives poor estimation if there is large bearing angle error or if there is any sort of discontinuity in the measurements, in both the cases there is a chance of violating assumption of local linearity.

A technique named Derivative Free Kalman Filter (DFKF) [2] helps to alleviate the problems associated with EKF. This filter yields an identical performance as compared to EKF when the assumption of local linearity is not violated. The DFKF operates on the principle that it is easier to approximate a Gaussian distribution than it is to approximate (i.e. linearization) an arbitrary non-linear function. It does not require any sort of linearization and uses deterministic sampling approach to capture the mean and covariance estimates with a minimal set of sample points or so called sigma points. The emphasis is shifted from linearization of non-linear system to sampling approach of probability density function.

In this paper, two schemes: factorized version of EKF and DFKF are studied and evaluated using various sets of simulated data corresponding to the non-linear systems as well as one real data set. In addition to that, sensitivity study of DFKF with respect to tuning parameters such as α, β, and κ (used in creation of sigma points and their associated weights) is also carried out using one set of simulated data. The concept of DFKF is extended to data fusion for similar sensors. Application of DFKF is demonstrated in parameter estimation problem also.

General Representation of Non-linear Systems

Consider a non-linear system/model in discrete domain:

$$X(k+1) = f\,[X(k), u(k), w(k), k] \tag{1}$$

where $X(k)$ is the n-dimensional state of the model, $u(k)$ is control input vector, n-dimensional $w(k)$ is the process noise vector (due to external disturbances or/and modeling errors) with Gaussian distribution and zero mean with covariance matrix Q, and k is the scan number. Assume that some of the states of the system represented by Eq.(1) are observed by sensor that has the following model:

$$Z_m(k) = h\,[X(k), u(k), k] + v(k) \tag{2}$$

where $Z_m(k)$ is the m-dimensional measurement vector, m-dimensional $v(k)$ is the measurement noise with Gaussian distribution and zero mean with a covariance matrix R.

Upper-Diagonal Extended Kalman Filter-UDEKF

The algorithm UDKF is numerically stable and computationally efficient method for implementing Kalman filter [1]. The term "U-D covariance factorization" comes from a property of non negative definite symmetric matrices, according to which such a matrix P can be factored into $P = UDU^T$, where U is an upper triangular matrix with unit elements on its main diagonal and D is a diagonal matrix. The algorithm for this filter is given in two parts i.e. i) state and covariance propagation and ii) state and covariance update. In present paper, UDEKF is used for the comparison with DFKF. Due to page limits, UDEKF equations can be found in [4].

Derivative Free Transformation and DFKF

As far as numerical stability is concerned, UDEKF provides an edge over conventional EKF, but the basic limitations of EKF remain same even for UDEKF. Therefore, it becomes very essential to have an algorithm that is possibly more accurate than linearization and meets the requirements of almost similar implementation and computational cost as of EKF.

The fundamental difference behind the working principle of EKF and DFKF [2] is that, in EKF, the non-linear models are linearised to parameterise the probability density function (pdf) in terms of its mean and covariance, whereas, for DFKF linearization is avoided and pdf is parameterised through non-linear transformation of deterministically chosen sample points. The non-linear transformation is termed as derivative free transformation (DFT) due to a fact that the transformation does not involve any sort of derivative expression. The DFT technique is firmly established on the fact that *it is easier to approximate a Gaussian distribution than to approximate an arbitrary non-linear function or transformation*. Fig. 1 shows pictorial representation of DFT for the following example:

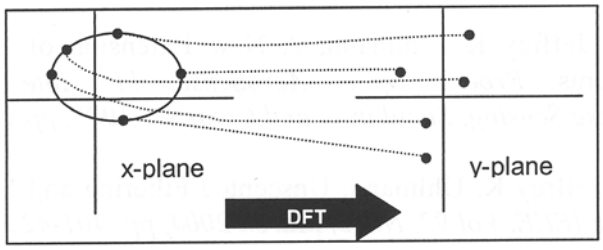

Fig.1 Derivative fre transformation [3]

Consider propagation of a random variable x of dimension L (in Fig.1, L=2) through a non-linear function $y = f(x)$. Assume that mean and covariance of sigma points, shown by black dots in left side of Fig.1, for random variable are $\bar{x}$ and P_x respectively. These sigma points and their associated weights are deterministically created using the following equations:

$$\begin{aligned}
\chi_0 &= \bar{x} \\
\chi_i &= \bar{x} + (\sqrt{(L+\gamma)\,P_x})_i \qquad i = 1, \ldots, L \\
\chi_i &= \bar{x} - (\sqrt{(L+\gamma)\,P_x})_{i-L} \qquad i = L+1, \ldots, 2L
\end{aligned} \tag{3}$$

$$\begin{aligned}
W_0^{(m)} &= \frac{\lambda}{L+\lambda} \\
W_0^{(c)} &= \frac{\lambda}{L+\lambda} + (1-\alpha^2+\beta) \\
W_i^{(m)} &= W_i^{(c)} = \frac{I}{2(L+\lambda)} \qquad i = 1, \ldots, 2L
\end{aligned} \tag{4}$$

The associated weights can be positive or negative, but to provide unbiased transformation, they must satisfy the condition $\sum_{i=1}^{2L} W_i^{(m\ or\ c)} = 1$. For square root in Eq. (3), it is proposed to use numerically efficient and stable method such as Cholesky decomposition. The scaling parameters used for the creation of sigma points and their associated weights are: α which determines the spread of sigma points around $\bar{x}$, β is used to incorporate any prior knowledge about distribution of $\bar{x}$, $\lambda = \alpha^2 (L+\kappa) - L$ and κ is the secondary tuning parameter. The sigma points created using Eq. (3) are propagated through the non-linear function ($y_i = f(\chi_i)$, where, $i = 0, ..., 2L$) resulting in transformed sigma points (black dots in right side of Fig.1). The mean and covariance of transformed points are formulated as:

$$\bar{y} = \sum_{i=0}^{2L} W_i^{(m)} y_i \tag{5}$$

$$P_y = \sum_{i=0}^{2L} W_i^{(c)} \{y_i - \bar{y}\}\{y_i - \bar{y}\}^T \tag{6}$$

The DFKF is straightforward extension of the DFT for the recursive estimation problems. The state of DFKF can be reconstructed by introducing the concept of augmented state vector that consists of actual system and process noise states each with n-dimension, and m-dimensional measurement noise state. Then the dimension of augmented state vector becomes $na = n+n+m = 2n+m$. Although augmentation technique lands up in use of additional sigma points, it implicitly incorporates the effects noises at various stages. The steps required in implementation of DFKF at every sampling point can be specified as follows in some details:

DFKF Initialization

$$\begin{aligned}
\hat{X}(0/0) &= E[X(0/0)] \\
\hat{P}(0/0) &= E\left[\left(X(0/0) - \hat{X}(0/0)\right)\left(X(0/0) - \hat{X}(0/0)\right)^T\right]
\end{aligned} \tag{7}$$

Augmented state and its error covariance are represented as :

$$\hat{X}^a(0/0) = E\left[X^a(0/0)\right]$$

$$= \left[X^T(0/0) \quad \underset{n-dim\ w}{0, \ldots, 0} \quad \underset{m-dim\ v}{0, \ldots, 0}\right]^T$$

$$\hat{P}^a(0/0) = E\left[\left(X^a(0/0) - \hat{X}(0/0)\right)\left(X^a(0/0) - \hat{X}^a(0/0)\right)^T\right]$$

$$= \begin{bmatrix} \hat{P}(0/0) & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & R \end{bmatrix}_{2n+m \ by \ 2n+m} \quad (8)$$

Sigma Points Computation

$$\chi_0^a(k/k) = \hat{X}^a(k/k)$$

$$\chi_i^a(k/k) = \hat{X}^a(k/k) + \left(\sqrt{(n_a+\lambda)\,\hat{P}^a(k/k)}\right)_i \quad i = 1, \ldots, n_a$$

$$\chi_i^a(k/k) = \hat{X}^a(k/k) - \left(\sqrt{(n_a+\lambda)\,\hat{P}^a(k/k)}\right)_{i-n_a} \quad i = n_a+1, \ldots, 2n_a \quad (9)$$

where, $\chi^a = \begin{bmatrix} \chi & \chi^w & \chi^v \\ - & - & - \\ state & process\ noise & meas.\ noise \end{bmatrix}$

State and Covariance Propagation

$$\chi(k+1/k) = f\left(\chi(k/k), u(k) + \chi^w(k/k), k\right)$$

$$\tilde{X}(k+1/k) = \sum_{i=0}^{2n_a} W_i^{(m)}\, \chi_i(k+1/k)$$

$$\tilde{P}(k+1/k) = \sum_{i=0}^{2n_a} W_i^{(c)} \left[\chi_i(k+1/k) - \tilde{X}(k+1/k)\right]$$

$$\left[\chi_i(k+1/k) - \tilde{X}(k+1/k)\right]^T \quad (10)$$

$$W_0^{(m)} = \frac{\lambda}{n_a+\lambda}$$

$$W_0^{(c)} = \frac{\lambda}{n_a+\lambda} + (1 - \alpha^2 + \beta)$$

$$W_i^{(m)} = W_i^{(c)} = \frac{1}{2(n_a+\lambda)} \quad i = 1, \ldots, 2n_a \quad (11)$$

State and Covariance Update

$$y(k+1/k) = h(\chi(k/k), u(k), k) + \chi^v(k/k)$$

$$\tilde{Z}(k+1/k) = \sum_{i=0}^{2n_a} W_i^{(m)}\, y_i(k+1/k) \quad (12)$$

$$S = \sum_{i=0}^{2n_a} W_i^{(c)} \left[y_i(k+1/k) - \tilde{Z}(k+1/k)\right]$$

$$\left[y_i(k+1/k) - \tilde{Z}(k+1/k)\right]^T$$

$$P_{xy} = \sum_{i=0}^{2n_a} W_i^{(c)} \left[\chi_i(k+1/k) - \tilde{X}(k+1/k)\right]$$

$$\left[y_i(k+1/k) - \tilde{Z}(k+1/k)\right]^T$$

$$K = P_{xy} S^{-1} \quad (filter\ gain) \quad (13)$$

$$\hat{X}(k+1/k+1) = \tilde{X}(k+1/k) + K\left(Z_m(k+1) - \tilde{Z}(k+1/k)\right)$$

$$\hat{P}(k+1/k+1) = \tilde{P}(k+1/k) - K S K^T \quad (14)$$

Results of Numerical Simulation

Example 1: Target Tracking

A 3D Cartesian simulation is carried out for an aerospace vehicle moving with constant acceleration and locked by a sensor capable of giving vehicle information in terms of range (meter), azimuth (rad.), and elevation (rad.). For the simulation of process model and sensor model, the following information is used:

- true (actual) initial state of the vehicle:

$X_t(0/) = [x \ \dot{x} \ \ddot{x} \ y \ \dot{y} \ \ddot{y} \ z \ \dot{z} \ \ddot{z}]$

$= [10 \ \ 10 \ \ 0.1 \ \ 10 \ \ 5 \ \ 0.1 \ \ 1000 \ \ 0 \ \ 0]$

- sampling interval (T) and total flight time (T_F) : 0.1 sec and 100 sec (*note that simulation is carried out with fixed-step size to save memory and to speed up process which may not be possible with variable step-size*)

- process noise variabce : Q = 0.01

- process mode (F) :

$$F1 = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}; \quad F = \begin{bmatrix} F1 & 0 & 0 \\ 0 & F1 & 0 \\ 0 & 0 & F1 \end{bmatrix}$$

- process noise matrix (G) :

$$G1 = \begin{bmatrix} \frac{T^3}{6} & 0 & 0 \\ 0 & \frac{T^2}{2} & 0 \\ 0 & 0 & T \end{bmatrix}; \quad G = \begin{bmatrix} G1 & 0 & 0 \\ 0 & G1 & 0 \\ 0 & 0 & G1 \end{bmatrix}$$

The noisy polar measurements are generated using following model :

$$Z_m\}(k) = \begin{bmatrix} r(k) & \theta(k) & \phi(k) \\ - & - & - \\ range & azimuth & elevation \end{bmatrix}$$

$$r(k) = \sqrt{x(k)^2 + y(k)^2 + z(k)^2} + n_r(k)$$

$$\theta(k) = \tan^{-1}(y(k)/x(k)) + n_\theta(k)$$

$$\phi(k) = \tan^{-1}(z(k)/\sqrt{x(k)^2 + y(k)^2}) + n_\phi(k) \qquad (15)$$

Here, n_r , n_θ and n_ϕ are random noise sequences with Gaussian distribution. The standard deviation of measurement noise for range (σ_r), azimuth (σ_θ) and elevation are computed offline based on pre-specified Signal-to-Noise Ratio (SNR) of 10. The measurement noise covariance matrix R can be written as

$$R = \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_\theta^2 & 0 \\ 0 & 0 & \sigma_\phi^2 \end{bmatrix}$$

The state estimation process is carried out using UDEKF and DFKF algorithms.

UDEKF and DFKF Initialization

For UDEKF, its initial state $\hat{X}^1(0/0)$ is kept near to $X_t(0/0)$, initial UD matrix contains

$$\hat{P}^1(0/0) = E\left[(X_t(0/0) - \hat{X}^1(0/0)\,(X_t(0/0) - \hat{X}^1(0/0))^T\right]$$

and, G, i.e., $\left[\hat{P}^1(0/0) \mid G\right]$. In case of DFKF, its initial state $\hat{X}^2(0/0)$ and its covariance $\hat{P}^2(0/0)$ are kept equal to $\hat{X}^1(0/0)$ and

$$\begin{bmatrix} \hat{P}^1(0/0) & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & R \end{bmatrix}$$ respectively. The suitablevalues after various trials and run of additional tuning parameters (for DFKF only) such as α, β, and κ are found to be 1, 0, and 0 respectively. There is no straight rule or methodology to find out the value of the tuning parameters and in present case they are chosen based on observing the states error (true-minus estimated) becoming minimum (nearly optimal!). The results are generated for 25 Monte Carlo simulations (for every run, seed of random number generator, used to generate measurement noise, is changed) only since no significant change in results is noticed for higher number (more than 25) of Monte Carlo runs. In Fig. 2, the true and estimated target states in x-axis are compared for both the filters. From this figure (especially by seeing velocity and acceleration states), it is clear that estimated states from DFKF are closer to true values than that from UDEKF. Also DFKF shows less estimation lag (approximately 7 second less) compared to UDEKF. Additionally, both filters are compared in term of Root Sum Square Position Error

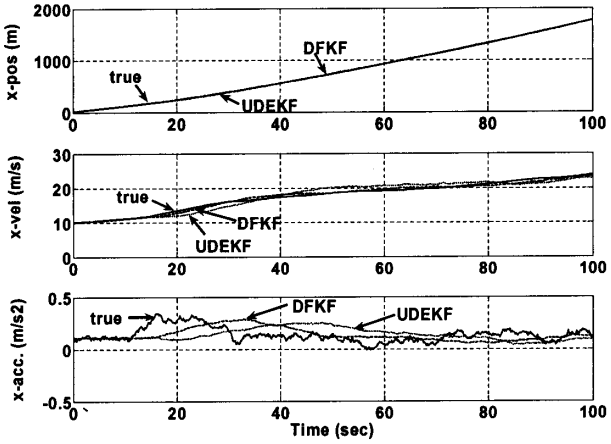

Fig.2 Comparison of true and estimated states - Example 1

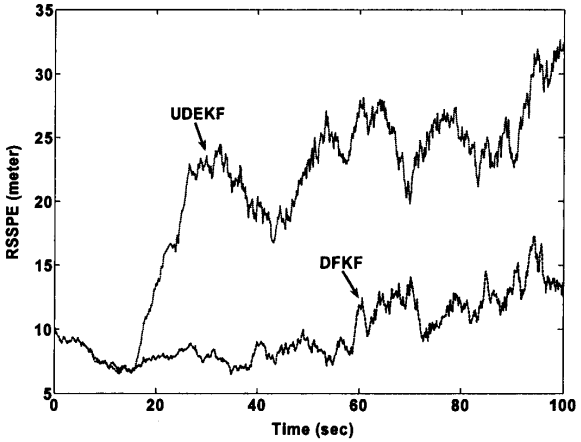

Fig.3 RSSPE calculated for UDEKF and DFKF - Example 1

$$\text{RSSPE} = \frac{1}{25}\sum_{m=1}^{25}\sqrt{(x_t(k)-\hat{x}(m,k))^2+(y_t(k)-\hat{y}(m,k))^2+(z_t(k)-\hat{z}(m,k))^2}\ ;$$

$k = 1, ..., N$,

where N is the total data point, $\hat{x}$, $\hat{y}$ and $\hat{z}$ are the estimated target positions in x, y and z axis respectively, and m is the Monte Carlo run index. The results are shown in Fig. 3. It can be observed that the error due to UDEKF is initially comparable with DFKF but increases with time whereas DFKF shows almost constant error, much lesser in magnitude for entire scans.

Example 2: Kinematic Consistency

In second example, an effort has been made to check the performance of both the filters for kinematics consistency (required for flight path reconstruction) using realistic longitudinal short period and lateral-directional (for flight condition: Mach = 0.5 and altitude = 4 Km), generated from a six-degree-of freedom simulation of an aircraft. The basic kinematic models required in state estimation are as follows:

$$\dot{u} = -(q-\Delta q)\,w + (r-\Delta r)\,v - g\sin\theta + (a_x - \Delta a_x),$$

$$\dot{v} = -(r-\Delta r)\,u + (p-\Delta p)\,w + g\cos\theta\sin\phi + a_y,$$

$$\dot{w} = -(p-\Delta p)\,v + (q-\Delta q)\,u + g\cos\theta\cos\phi + (a_z - \Delta a_z),$$

$$\dot{\phi} = (p-\Delta p) + (q-\Delta q)\sin\phi\tan\theta + (r-\Delta r)\cos\phi\tan\theta,$$

$$\dot{\theta} = (q-\Delta q)\cos\phi - (r-\Delta r)\sin\phi,$$

$$\dot{h} = u\sin\theta - v\cos\theta\sin\phi - w\cos\theta\cos\phi \qquad (16)$$

where, Δa_x, Δa_z, Δp, Δq, Δr, K_α, K_θ are the bias terms, and p, q, r, a_x, a_y, a_z are the control inputs to the process model.

Observation or Measurement Model

$$Z_{t|m} = \begin{bmatrix} V_m & \alpha_m & \beta_m & \phi_m & \theta_m & h_m \end{bmatrix}$$

$$V_m = \sqrt{u_n^2 + v_n^2 + w_n^2}, \quad \alpha_m = K_\alpha \tan^{-1}\left[\frac{w_n}{u_n}\right],$$

$$\beta_m = \sin^{-1}\left[\frac{v_n}{\sqrt{u_n^2 + v_n^2 + w_n^2}}\right],$$

$$\phi_m = \phi + \Delta\phi, \quad \theta_m = K_\theta\,\theta, \quad h_m = h \qquad (17)$$

where, u_n, v_n, w_n are the velocity components along the three axes at the nose boom of the aircraft and computed as follows:

$$u_n = u - (r-\Delta r)\,Y_n + (q-\Delta q)\,Z_n,$$
$$v_n = v - (p-\Delta p)\,Z_n + (r-\Delta r)\,X_n,$$
$$w_n = w - (q-\Delta q)\,X_n + (p-\Delta p)\,Y_n \qquad (18)$$

where, X_n, Y_n and Z_n are the offset distances from nose boom to CG and their values are kept at 12.586, 0.011, and 0.14 respectively. The measurement noise with SNR of 10 is added externally only to the observables V, α, β, ϕ, θ, h. No noise is added to the rates and accel-

erations during the data generation because in general rate and acceleration sensors are supposed to be very accurate.

The additional information used in both the filters is:

- initial state $\hat{X}^1(0/0)$ and $\hat{X}^2(0/0)$

$= [u\ v\ w\ \phi\ \theta\ h\ \Delta a_x\ \Delta a_z\ \Delta p\ \Delta q\ \Delta r\ K_\alpha\ K_\theta]$

$= [167\ \ 0.001\ \ 17.305\ \ 0\ \ 0.10384\ \ 4000\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 1]$

- sampling interval : T = 0.025 sec. and process noise variance : Q = 1.0e - 15* (*eye (nx*)

- measurement noise variance :

$R = E\left[\left(Z_m - Z_t\right)\left(Z_m - Z_t\right)^T\right]$ where Z_t is the noise free measurement from simulator and Z_m is the noisy measurement

- initial state errror covariance : *eye* (*nx*) for UDEKF and and

$$\begin{bmatrix} eye(nx) & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & R \end{bmatrix}$$ for DFKF.

where, *nx* = 13 is the number of estimated states.

The results shown in Figs. 4-5 are generated for 25 Monte Carlo simulations. Fig. 4 shows the comparison of true, measured and estimated observables such as V, α, β, ϕ, θ, h. From the plots it is clear that wherever

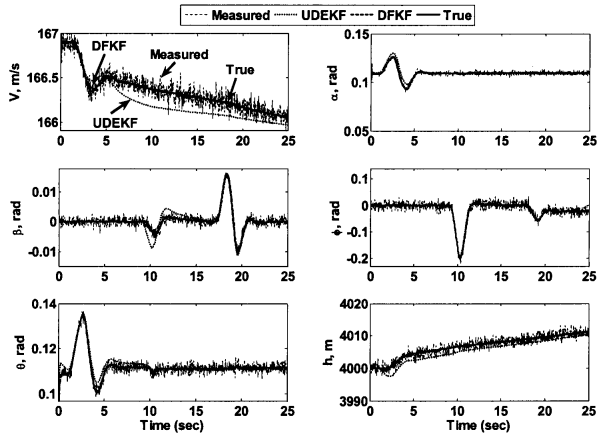

Fig.4 Comparison of true, measured, and estimated observation date - Example 2

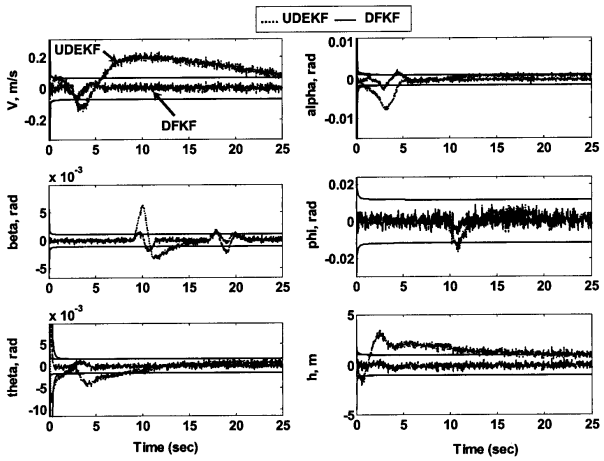

Fig.5 Comparison of innovation sequence with (1σ) bounds - Example 2

(e.g. between 0-5 seconds or around 10 second) the non-linearity in measurement data is more severe, the performance of UDEKF is degraded as compared to DFKF. This can be further proved by comparing the measurement residuals or innovation sequences with corresponding 1σ ($\sigma = \pm\sqrt{HPH' + R}$) theoretical bounds which represent the bounds on the accuracies achievable by the filter given the covariances (P, Q, R). If 1σ bounds are plotted, theoretically it is expected that for ideal filter performance, 90% of the innovation sequence points should be within bounds.

It is observed from Fig. 5 that the bounds are comparable (due to same initial conditions) for both filters but in case of UDEKF its residuals go out of bounds for more number of times as compared with DFKF.

Example 3: Sensitivity Study of DFKF

The common tuning parameters for UDEKF and DFKF are initial state covariance (P_0), process noise variance (Q) and measurement noise variance (R). The affects of these three parameters on the filters performance are well known. The additional tuning parameters (not required/specified for UDEKF) in DFKF areα, β, and κ. The sensitivity study of DFKF is carried out using these additional parameters only.

For this example, a two-dimensional particle is considered with constant speed in x-direction only. It is initially released in x-direction [3]. The goal is to estimate the mean and covariance of target state for rest of its projectile. The discontinuities in system dynamics are introduced by ob-

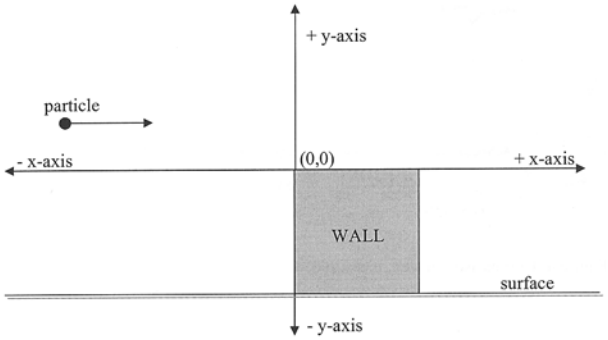

Fig.6 Pictorial representation of a discontinuous system [3] - Example 3

structing the path of particle by a wall located in the "bottom right quarter plane" ($x \geq 0, y \leq 0$), see Fig.6. If the particle hits the wall, with an assumption of perfectly elastic collision, its projectile is reflected back at the same velocity as it traveled forward. The process model for the particle at any instant of time is represented by following equations:

$$x(k+1) = \begin{cases} x(k) + T\dot{x}(k) & y(k) \geq \\ -(x(k) + T\dot{x}(k)) & y(k) < 0 \end{cases}$$

$$\dot{x}(k+1) = \dot{x}(k)$$

$$y(k+1) = y(k) + w(k)$$

$$\dot{y}(k+1) = \dot{y}(k) \qquad (19)$$

where, T is the sampling interval, k is the scan number, and w is the process noise with Gaussian distribution and has zero mean with variance of Q. In present case, initial state of target is kept at $x = -10$, $\dot{x} = 1$, $y = 0$, $\dot{y} = 0$. Here, $N = 20$, $T = 0.1$ and $Q = 0.001$. The particle positions in x and y directions are measured by sensor placed at origin of x-y plane and its noise variance R is equal to $\begin{bmatrix} 1 & 0 \\ 0 & 0.001 \end{bmatrix}$. The state estimation is carried out using DFKF with its initial state 0.01% less than true initial value and state error covariance

$$P_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The results are generated for 100 Monte Carlo simulations for various cases (for sensitivity study) obtained by appropriate selection of tuning parameters such as α, β, and κ. The sensitivity study is carried out for following cases:

Parameter →	α	β	κ
case 1	varying	0	0
case 2	1	0	varying
case 3	1	varying	0

In order to retain the positive semi definiteness of predicted covariance matrix (refer Eq. (10)), a sensitivity study has been carried out on positive values of tuning parameters only. From various trials and run it is found that DFKF gives optimal estimation for $\alpha = 1$, $\beta = 0$, $\kappa = 0$. Keeping this as a reference values (for performance comparison), above three cases are created. In case 1, estimation is carried out for $\alpha = 0.1, 0.5, 1, 5, 10$. Fig.7 compares the RSSPE for various values of α. From the figure following observations are made:

- For low or high α, RSSPE is high
- For extreme low and extreme high α, RSSPE are comparable
- For α - > 1, RSSPE approaches to an optimal value (when $\alpha = 1$, $\beta = 0$, $\kappa = 0$)
- RSSPE is highly sensitive to α

In case 2, estimation is carried out for $\kappa = 0, 1, 5, 10$. Fig. 8 compares the RSSPE for various values of κ. From the figure following observations are made :

- with increase in κ, RSSPE increases
- For κ - > 0, RSSPE approaches to an optimal value
- RSSPE is moderately sensitive to κ

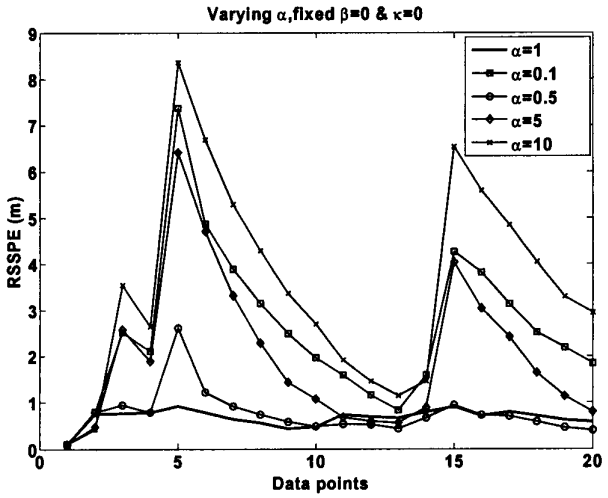

Fig.7 RSSPE calculated for different values of α - Example 3

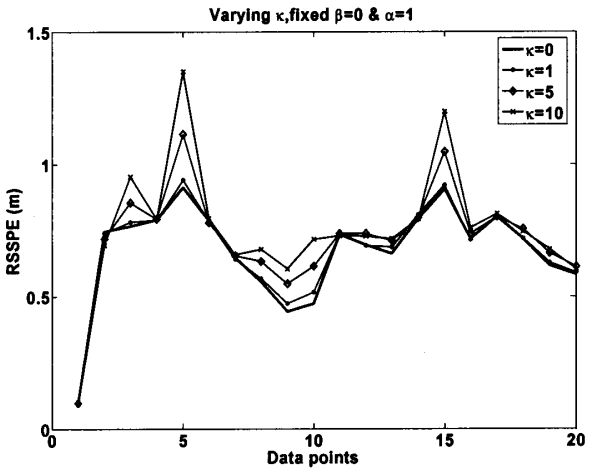

Fig.8 RSSPE calculated for different values of κ- *Example 3*

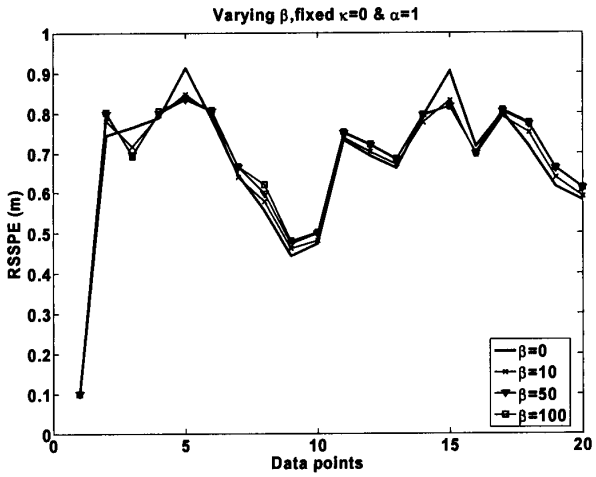

Fig.9 RSSPE calculated for different values of β- *Example 3*

In case 3, estimation is carried out for b = 0. 10. 50, 100. Fig. 9 compares the RSSPE for various values of β. From the figure following observations are made :

- with increase in β, RSSPE fluctuates around an optimal value
- For β - > 0, RSSPE approaches to an optimal value
- RSSPE is less sensitive to β

In summary, i) estimation performance of DFKF is more sensitive to tuning parameter α. The reason could be that α is the key parameter which decides the spread of sigma points around mean value of estimated state, ii) filter is moderately sensitive to κ and less sensitive to secondary tuning parameter β, and iii) parameter β can be used for fine tuning to reduce the overall prediction errors thereby enhancing the filter performance to some extent.

Example 4: Parameter Estimation

DFKF algorithm is applied to perform estimation of non-dimensional longitudinal parameters using simulated short period data of an aircraft. The relevant mass, moment of inertia and other aircraft geometry related parameters are provided below:

Mass	=	2280.0 kg
Moment of inertia, Iyy	=	6940.0 kg-m^2
Mean aerodynamic chord, $\bar{c}$	=	1.5875 m
Wing area, S	=	23.23 m^2
Air density, ρ	=	0.9077 kg/m^3
u_0 = 36 m/s and w_0 = 7.2 m/s		

The data is simulated with a sampling time of 0.03 second by giving a doublet input to the elevator. Random process noise (zero mean and Gaussian) with standard deviation of 0.001 is added to certain states u, w, q, and θ. The noisy measurements u, w, q, θ, a_x, a_z and $\dot{q}$ with SNR=10 are generated. The state and measurement models for estimation of the parameters in body axis are formulated as follows:

State Model

$$\dot{u} = \frac{\bar{q}S}{m} C_X - qw - g \sin\theta$$

$$\dot{w} = \frac{\bar{q}S}{m} C_Z - qu - g \cos\theta$$

$$\dot{q} = \frac{\bar{q}S\,\bar{c}}{I_{yy}} C_m$$

$$\dot{\theta} = q \qquad (20)$$

where,

$$C_Z = C_{Z_0} + C_{Z_\alpha}\,\alpha + (-4.32)\frac{q\,\bar{c}}{2\,V} + C_{Z_{\delta_e}}\,\delta_e$$

$$C_X = C_{X_0} + C_{X_\alpha}\,\alpha + C_{X_{\alpha^2}}\,\alpha^2$$

$$C_m = C_{m_0} + C_{m_\alpha}\,\alpha_m + C_{m_{\alpha^2}}\,\alpha^2 + C_{m_q}\frac{q\,m\,\bar{c}}{2\,V} + C_{m_{\delta_e}}\,\delta_e \qquad (21)$$

where,

$$\bar{q} = 0.5\,\rho\,V^2,\ V = \sqrt{u^2 + w^2} \text{ and } \alpha = \tan^{-1}(w/u)$$

Measurement Model

$$Z_{t|m} = \begin{bmatrix} y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 \end{bmatrix}$$

$$y_1 = u\,; \qquad y_2 = w$$

$$y_3 = q\,; \qquad y_4 = \theta$$

$$y_5(a_x) = \frac{\bar{q}\,S}{m} C_X\,; \quad y_6(a_z) = \frac{\bar{q}\,S}{m} C_Z$$

$$y_7 - \bar{q} \tag{22}$$

For estimating the parameters $C_{()}$ i.e. RHS of Eq. (21) using DFKF, they are modeled as augmented states in the state model specified by Eq. (20). In this case there are 4 states and 11 parameters to be estimated using 7 observables specified by Eq. (22). The initial states and parameters for the DFKF are assumed to be 10% off from their true values. The initial estimation covariance matrix is chosen to reflect this uncertainty. The tuning parameters Q and R are as follows:

$$Q = diag\,[1.0e-6 \quad 1.0e-6 \quad 1.0e-6 \quad 1.0e-6]\,,$$

$$R = E\left[\left(Z_m - Z_t\right)\left((Z_m - Z_t)\right)^T\right],$$

where, Z_m and Z_t stand for noisy and clean measurements respectively. The additional tuning parameters such as α, β, and κ are kept at 1, 0, and 0 respectively.

The estimated values of the parameters are compared with the true values of the derivatives in Table-1. The estimates are fairly close to the true values. The convergence of the pitching moment related derivatives: C_{m_α}, $C_{m_{\alpha^2}}$, C_{m_q}, $C_{m_{\delta_e}}$ is shown in Fig. 10. It is clear that even in the presence of noise in the data, the parameters converge close to their true values. However, some deviation is observed for C_{m_q} estimate.

Example 5: Target Tracking using Real Data

The performance of DFKF is evaluated using real data. The real data consists of target information in polar frame measured by sensor. The state estimation, consisting of targets position, velocity, and acceleration information, is carried out in 3D Cartesian frame using constant acceleration model (mentioned in example 1) and polar data. The sampling time interval chosen is 0.25 second with process

Table-1 : Estimated parameters of an aircraft - Example-4

Parameter	True Values	Estimated
C_{X_0}	-0.9540	-0.0616
C_{X_a}	0.2330	0.2531
$C_{X_a^2}$	3.6089	3.6840
C_{Z_0}	-0.1200	-0.1279
C_{Z_a}	-5.6800	-5.7084
$C_{Z_{\delta_e}}$	-0.4070	-0.5033
C_{m_0}	0.0550	0.0604
C_{m_a}	-0.7290	-0.7108
$C_{m_a^2}$	-1.7150	-1.7701
C_{m_q}	-16.3	-14.9726
$C_{m_{\delta_e}}$	-1.9400	-1.8779

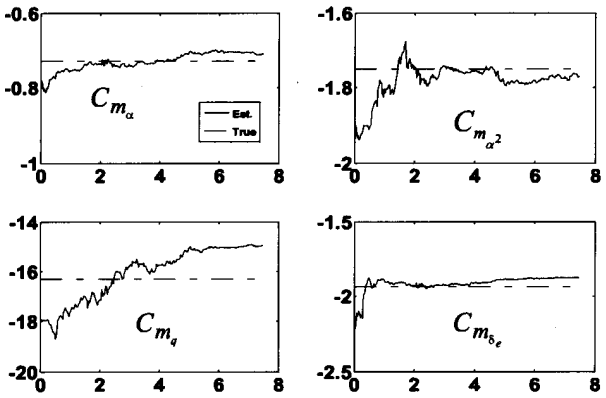

Fig.10 Parameter convergence for an aircraft-pitching moment derivatives - Example 4

noise covariance Q kept at 0.01*eye (9,9), whereas, measurement noise covariance value R is obtained from sensor specification. The initial state of DFKF is kept close to true initial target state and accordingly initial state error covariance is computed. The additional tuning parameters α, β, and κ are kept at 1, 0, and 0 respectively.

Figure 11 illustrates measured, true and estimated polar data. It is clear from the plots that estimated data is fairly close to true data. Some discontinuities (more clearly visible in range), due to sensor data loss, are observed somewhere between 5000-6000 data points. Fig.12 shows the RSSPE (definition mentioned in example 1) computed w.r.t. true target positions. It is noticed

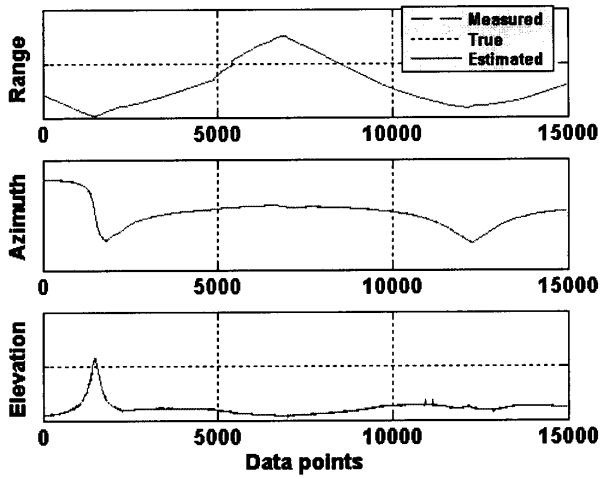

Fig.11 Comparison of meas., true, and estimated polar data - Example 5

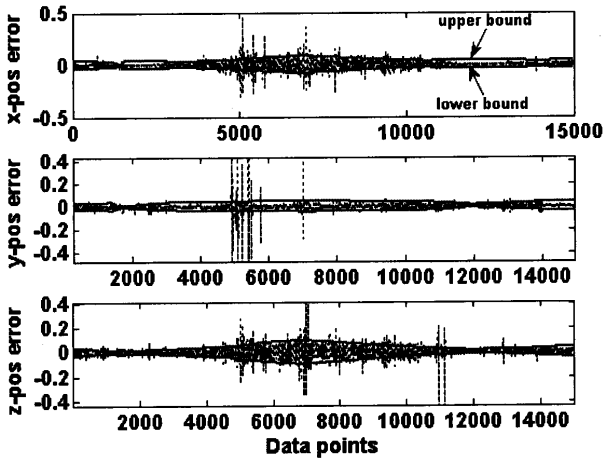

Fig.13 Target's position error compared with theoretical bounds - Example 5

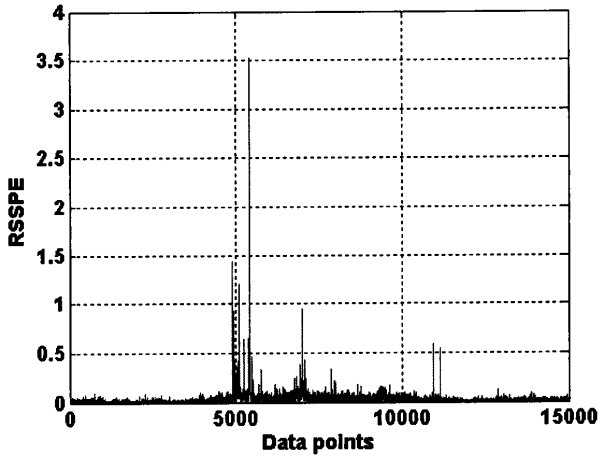

Fig.12 RSSPE w.r.t. true data - Example 5

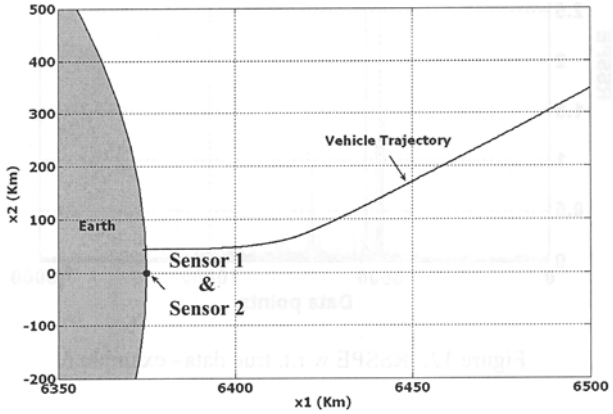

Fig.14 True positions of re-entered vehicle to earth atmosphere [3] - Example 6

from the figure that on average RSSPE value comes around 0.1 on y-scale and it is comparatively high only at those points where sensor data loss is encountered. Additionally, a sudden increase in RSSPE is attributed to spikes/outliers observed in sensor data. Fig.13 compares the state error (true position - estimated position) with 1 sigma bound i.e. $\pm\sqrt{diag(p)}$. It can be observed from the figure that most of the time state error does not exceed its theoretical bounds and where ever it exceeds the bounds is either due to data loss or spikes present in sensor data.

Example 6: Data Fusion

Consider the vehicle reentry problem [3,5] shown in Fig. 14. It is assumed that a vehicle entering the atmosphere at high altitude and at high speed is tracked by two ground-stationed sensors with different accuracies. It is assumed that sensors are placed nearby. The measurements from either of sensors are in terms of range and bearing. The strong non-linearities present in vehicle dynamic are due to the different types of forces acting on it. The most dominant force is aerodynamic drag as a function of vehicle speed and altitude. Gravitational force accelerates the vehicle towards the center of Earth. In initial phase of flight, vehicle has almost ballistic trajectory but as density of the atmosphere increases, drag effects become more important and the vehicle rapidly decelerates until its motion is almost vertical. The state space formulation of vehicle dynamics is as follows:

$$\dot{x}_1(k) = x_3(k)$$

$$\dot{x}_2(k) = x_4(k)$$

$$\dot{x}_3(k) = D(k)\,x_3(k) + G(k)\,x_1(k) + w_1(k)$$
$$\dot{x}_4(k) = D(k)\,x_4(k) + G(k)\,x_2(k) + w_2(k)$$
$$\dot{x}_5(k) = w_3(k) \qquad (23)$$

where, x_1 and x_2 are the target positions in 2-D plane, x_3 and x_4 are the corresponding velocities, x_5 is a parameter related to aerodynamic properties, is the drag-related term, is the gravity-related term and w_1, w_2, w_3 are uncorrelated white Gaussian process noises with zero mean and standard deviations of $\sigma_{w_1} = 0.0049$, $\sigma_{w_1} = 0.0049$ and $\sigma_{w_1} = 4.9e - 8$ respectively. In Eq. (23), drag and gravitational terms are computed using following equations.

$$D(k) = -\beta(k)\exp\left\{\frac{r_0 - r(k)}{H_0}\right\}V(k)$$
$$G(k) = -\frac{Gm_0}{r^3(k)}$$
$$\beta(k) = -\beta_0 \exp(x_5(k))$$
$$r(k) = \sqrt{x_1^2(k) + x_2^2(k)}$$
$$V(k) = \sqrt{x_3^2(k) + x_4^2(k)} \qquad (24)$$

where, $\beta_0 = -0.59784$, $H_0 = 13.406$, $G_{m0} = 3.9860 \times 10^5$ and $r_0 = 6374$ are the parameters reflecting typical environmental and vehicle characteristics. The initial state of vehicle is equal to [6500.4, 349.14, -1.8093, -6.7967, 0.6932]. The data is simulated for total number of N = 1450 scans. The vehicle is continuously tracked by two sensors in proximity at (x_r = 6375 Km, y_r = 0 Km). The rate at which measurements arrive is at a frequency of 5Hz i.e. sampling interval T = 0.2 seconds and model of sensor is represented by following equations:

$$r_i(k) = \sqrt{(x_1(k) - x_r)^2 + (x_2(k) - y_r)^2} + v_{ir}(k)$$
$$\theta_i(k) = \tan^{-}\left(\frac{x_2(k) - y_r}{x_1(k) - x_r}\right) + v_{i\theta}(k) \qquad (25)$$

where, r_i and θ_i are the measure range and bearing of i^{th} sensor, v_{ir} and $v_{i\theta}$ are the corresponding white Gaussian measurement noises.

It is assumed that first sensor gives good bearing information but has noisy range measurement and vice-versa for second sensor (thought this may not be true in general, it is assumed here for the sake of performance evaluation of the algorithm). The standard deviations of range and bearing noises used in simulation are:

Sensor 1 : σ_{1r} = 1 Km, $\sigma_{1\theta}$ = 0.05 deg
Sensor 2 : σ_{2r} = 0.22 Km, $\sigma_{2\theta}$ = 1 deg

In this paper, an effort has been made to evolve a Data Fusion (DF) scheme for similar sensors using DFKF to get more information about an entity of interest that would have been not possible by single sensor alone. In order to develop a fusion scheme, following assumptions and changes are required in DFKF algorithm formulated by Eqs. (7)-(14) :

Assumptions

- sensors are of similar type i.e. same data type/format
- measurements originating from sensors are synchronized in time

DF-DFKF Initialization

$$\hat{X}(0/0) = E[X(0/0)]$$
$$\hat{P}(0/0) = E\left[\left(X(0/0) - \hat{X}(0/0)\right)\left(X(0/0) - \hat{X}(0/0)\right)^T\right] \qquad (26)$$

Augmented state and its error covariance are presented as :

$$\hat{X}^a(0/0) = E\left[X^a(0/0)\right]$$
$$= \left[\hat{X}^T(0/0)\ \underbrace{0, \ldots, 0}_{n-dim\ w}\ \underbrace{0, \ldots, 0}_{m-dim\ v_1(sensors\ 1)}\ \underbrace{0, \ldots, 0}_{m-dim\ v_2(sensors\ 2)}\ 0, \ldots, 0\ \underbrace{0, \ldots, 0}_{m-dim\ v_{NS}(sensors\ NS)}\right]^T$$
$$\hat{P}^a(0/0) = E\left[\left(X^a(0/0) - \hat{X}^a(0/0)\right)\left(X^a(0/0) - \hat{X}^a(0/0)\right)^T\right]$$

$$= \begin{bmatrix} \hat{P}(0/0) & 0 & 0 & 0 & 0 & 0 \\ 0 & Q & 0 & 0 & 0 & 0 \\ 0 & 0 & R_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & R_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & ,..., & 0 \\ 0 & 0 & 0 & 0 & 0 & R_{NS} \end{bmatrix}_{2N+NS^{*}m \text{ by } 2n+NS^{*}m} \tag{27}$$

where, NS is the total number of sensors. The dimension of new augmented state vector becomes $n_a = n + n + NS^* m = 2n + NS^* m$.

Sigma Points Computation

Same as Eq. (9).

where,

$$\chi^a = \left[\underbrace{\chi}_{\text{state}} \quad \underbrace{\chi^{w}}_{\text{process noise}} \quad \underbrace{\chi^{v_1}}_{\text{meas. noise(sensor 1)}} \quad \underbrace{\chi^{v_2}}_{\text{meas. noise(sensor 2)}} \quad ,..., \quad \underbrace{\chi^{v_{NS}}}_{\text{meas. noise(sensor NS)}} \right]$$

State and Covariance Propagation

Same as Eqns (10) - (11).

State and Covariance Update

$$y^j(k+1/k) = h(\chi(k/k), u(k) + \chi^{v_j}(k/k)$$

$$\tilde{Z}^j(k+1/k) = \sum_{i=0}^{2n_a} W_i^{(m)} y_i^j(k+1/k) \tag{28}$$

where, j = 1, . . . , NS and

$$y(k+1/k) = \left[y^1(k+1/k) \;\; y^2(k+1/k), \ldots, \; y^{NS}(k+1/k) \right]^T$$

$$\tilde{Z}(k+1/k) = \left[\tilde{Z}^1(k+1/k) \;\; \tilde{Z}^2(k+1/k), \ldots, \; \tilde{Z}^{NS}(k+1/k) \right]^T$$

DF-DFKF uses Eqs.(13) - (14) for remaining computations. In Eq. (14), variable Z_m is reformulated as follows :

$$Z_m(k+1) = \left[Z_m^1(k+1) \;\; Z_m^2(k+1), \ldots, \; Z_m^{NS}(k+1) \right]^T$$

where, Z_m^1 , Z_m^2 , . . . , and Z_m^{NS} are the measuremnts from sensor 1, sensor 2, . . . , and sensor NS resepctively at k+1 scan.

The results are generated for 25 Monte Carlo simulations and performances of DF-DFKF and two DFKF (i.e. for sensor 1 and sensor 2 respectively) are compared. It is clear from Fig.15 that fused state, as compared to estimated state from other two methods, is close to true state. It is clear that data fusion increases the estimation accuracy that would not have been possible using single sensor measurements.

Conclusions

The performances of UDEKF and DFKF are compared for applications like target tracking using non-linear measurement model, and kinematic consistency checking using realistic longitudinal short period and lateral-directional data of an aircraft. It is observed that DFKF performs better than UDEKF and hence can be used for many non-linear filtering and control applications. Also, a sensitivity study of DFKF is carried out and found that the filter is more sensitive to tuning parameter α, moderately sensitive to κ and less sensitive to secondary tuning parameter β. Application of DFKF is also illustrated for parameter estimation. A data fusion scheme for similar sensors is proposed and its performance evaluated.

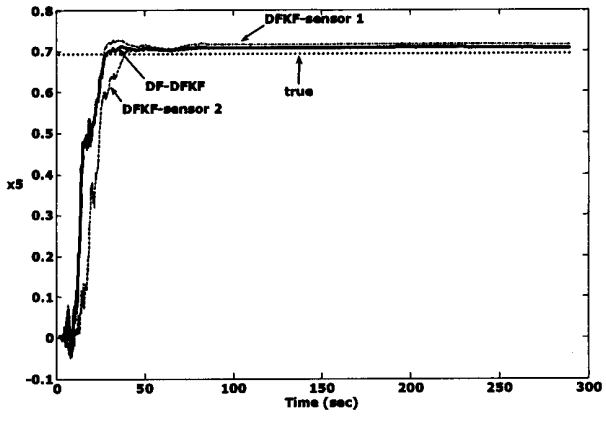

Fig.15 Comparison of true, estimated and fused state x_5 - Example 6

References

1. Gerald J. Bierman., "Factorization Methods for Discrete Sequential Estimation", Academic Press, New York, 1977.

2. Simon J. Julier and Jeffrey, K. Uhlmann., "A New Extension of the Kalman Filter to Non-linear Systems", Proceeding of AeroSense, 11th International Symposium Aerospace/Defense Sensing, Simulation and Controls, 1997, pp.182-193.

3. Simon J. Julier and Jeffrey K. Uhlmann., "Unscented Filtering and Non-linear Estimation", Proceeding of the IEEE, Vol.92, No. 3, March 2004, pp. 401-422.

4. Raol, J.R., Girija, G. and Jatinder Singh., "Modelling and Parameter Estimation of Dynamic Systems", IEE Control Engineering Series Book, Vol. No. 65, IEE, London, August 2004.

5. Kashyap, S.K. and Raol, J.R., "Evaluation of Derivative Free Kalman Filter and Fusion in Non-Linear Estimation", CCECE/CCGEI, IEEE Canadian Conference on Electrical and Computer Engineering, Ottawa, Canada, May 7-10, 2006.