16

Distributed Particle Filtering over Sensor Networks for Autonomous Navigation of UAVs

Gerasimos G. Rigatos
Industrial Systems Institute
Greece

1. Introduction

State estimation and control over sensor networks is a problem met in several applications such as surveillance and condition monitoring of large-scale systems, multi-robot systems and cooperating UAVs. In sensor networks the simplest kind of architecture is centralized. Distributed sensors send measurement data to a central processing unit which provides the state estimate for the monitored system. Such an approach has several weaknesses: (i) it lacks fault tolerance: if the central processing unit is subject to a fault then state estimation becomes impossible, (ii) communication overhead often prohibits proper functioning in case of a large number of distributed measurement units. On the other hand decentralized architectures are based on the communication between neighboring measurement units. This assures scalability for the network since the number of messages received or sent by each measurement unit is independent of the total number of measurement units in the system. It has been shown that scalable decentralized state estimation can be achieved for linear Gaussian models, when the measurements are linear functions of the state and the associated process and measurement noise models follow a Gaussian distribution (Nettleton et al. 2003). A solution to decentralized sensor fusion over sensor networks with the use of distributed Kalman Filtering has been proposed in (Olfati-Saber 2006), (Watanabe & Tzafestas 1992), (Olfati-Saber 2005), (Gan & Harris 2001), (Gao et al. 2009). Distributed state estimation in the case of non-Gaussian models has been studied in (Rosencrantz et al. 2003) where decentralized sensor fusion with the use of distributed particle filters has been proposed in several other research works (Mahler 2007), (Makarenko & Durrant-Whyte 2006), (Deming & Perlovsky 2007).

In this paper autonomous navigation of UAVs will be examined and a solution to this problem will be first attempted with the use of the Extended Information Filter and the Unscented Kalman filter (Shima et al. 2007), (Lee et al. 2008), (Lee et al. 2008), (Vercauteren & Wang 2005). Comparatively, autonomous UAV navigation with the use of the Distributed Particle Filter will be studied. This problem belongs to the wider area of multi-source multi-target tracking (Coué et al. 2006), (Hue et al. 2002), (Ing & Coates 2005), (Coué et al. 2003), (Morelande & D. Mušicki 2005). Subproblems to be solved for succeeding autonomous navigation of the UAVs are: (i) implementation of sensor fusion with the use of distributed filtering. In this approach the goal is to consistently combine the local particle distribution with the communicated particle distribution coming from particle filters running on nearby measurement stations (Caballero et al. 2008). It is assumed that each local measurement station runs its own local filter and communicates information to other measurement stations close to it. The motivation for using particle filters is that they can represent almost arbitrary probability distributions, thus becoming well-suited to accommodate the types of uncertainty and nonlinearities that arise in the distributed estimation (Rigatos 2009a), (Rigatos 2009b) (ii) nonlinear control of the UAVs based on the state estimates provided by the particle filtering algorithm. Various approaches have been proposed for the UAV navigation using nonlinear feedback control (Ren & Beard 2004), (Beard et al. 2002), (Singh & Fuller 2001). The paper proposes flatness-based control for the UAV models. Flatness-based control theory is based on the concept of differential flatness and has been successfully applied to several nonlinear dynamical systems. Flatness-based control for a UAV helicopter-like model has been developed in (Léchevin & Rabbath 2006), assuming that the UAV performs manoeuvres at a constant altitude.

The paper proposes first the Extended Information Filter (EIF) and the Unscented Information Filter (UIF) as possible approaches for fusing the state estimates provided by the local monitoring stations, under the assumption of Gaussian noises. The EIF and UIF estimated state vector is in turn used by a flatness-based controller that makes the UAV follow the desirable trajectory. The Extended Information Filter is a generalization of the Information Filter in which the local filters do not exchange raw measurements but send to an aggregation filter their local information matrices (local inverse covariance matrices) and their associated local information state vectors (products of the local information matrices with the local state vectors) (Shima et al. 2007), (Lee et al. 2008). In the case of the Unscented Information Filter there is no linearization of the UAVs observation equation. However the application of the Information Filter algorithm is possible through an implicit linearization which is performed by approximating the Jacobian matrix of the system's output equation by the product of the inverse of the state vector's covariance matrix (which can be also associated to the Fisher Information matrix) with the cross-correlation covariance matrix between the system's state vector and the system's output (Lee et al. 2008)], (Vercauteren & Wang 2005). Again, the local information matrices and the local information state vectors are transferred to an aggregation filter which produces the global estimation of the system's state vector.

Next, the Distributed Particle Filter (DPF) is proposed for fusing the state estimates provided by the local monitoring stations (local filters). The motivation for using DPF is that it is well-suited to accommodate non-Gaussian measurements. A difficulty in implementing distributed particle filtering is that particles from one particle set (which correspond to a local particle filter) do not have the same support (do not cover the same area and points on the samples space) as particles from another particle set (which are associated with another particle filter) (Ong et al. 2008), (Ong et al. 2006). This can be resolved by transforming the particles sets into Gaussian mixtures, and defining the global probability distribution on the common support set of the probability density functions associated with the local filters. The state vector which is estimated with the use of the DPF is used again by a flatness-based controller to make each UAV follow a desirable flight path.

The structure of the chapter is as follows: in Section 2 the Distributed Extended Kalman Filter (Extended Information Filter) is studied. In Section 3, the Distributed Unscented Kalman Filter (Unscented Information Filter) is analyzed and its use for distributed sensor fusion and state estimation is explained. In Section 4 Distributed Particle Filtering for sensor fusion-based state estimation will be analyzed. In Section 5 nonlinear control will be proposed for succeeding trajectory tracking by the UAVs. In Section 6 simulation experiments will be provided about UAVs autonomous navigation using the proposed distributed particle filtering algorithm. The test case will be concerned with m helicopter models monitored by n different ground stations. By fusing the measurements from the distributed observation units with the use of the Extended Information Filter and the proposed Particle Filter algorithm, state estimates of the UAVs are obtained. These in turn are used by local nonlinear controllers for succeeding trajectory tracking. Finally in Section 7 concluding remarks will be provided.

2. Distributed Extended Kalman Filtering

2.1 Extended Kalman Filtering at local processing units

The distributed Extended Kalman Filter, also know as Extended Information Filter, performs fusion of the state estimates which are provided by local Extended Kalman Filters. Thus, the functioning of the local Extended Kalman Filters should be analyzed first. The following nonlinear state model is considered (Rigatos & Tzafestas 2007):

$$\begin{aligned} x(k+1) &= \phi(x(k)) + L(k)u(k) + w(k) \\ z(k) &= \gamma(x(k)) + v(k) \end{aligned} \tag{1}$$

where $x \in R^{m\times 1}$ is the system's state vector and $z \in R^{p\times 1}$ is the system's output, while $w(k)$ and $v(k)$ are uncorrelated, zero-mean, Gaussian zero-mean noise processes with co- variance matrices $Q(k)$ and $R(k)$ respectively. The operators $\phi(x)$ and $\gamma(x)$ are $\phi\,(x) = [\phi_1(x), \phi_2(x), \cdots, \phi_m(x)]^T$, and $\gamma\,(x) = [\gamma_1(x), \gamma_2(x), \cdots, \gamma_p(x)]^T$, respectively. It is assumed that ϕ and γ are sufficiently smooth in x so that each one has a valid series Taylor expansion.

Following a linearization procedure, ϕ is expanded into Taylor series about $\hat{x}$:

$$\phi(x(k)) = \phi(\hat{x}(k)) + J_\phi(\hat{x}(k))[x(k) - \hat{x}(k)] + \cdots \tag{2}$$

where $J_\phi(x)$ is the Jacobian of ϕ calculated at $\hat{x}(k)$:

$$J_\phi(x) = \frac{\partial \phi}{\partial x}|_{x=\hat{x}(k)} = \begin{pmatrix} \frac{\partial \phi_1}{\partial x_1} & \frac{\partial \phi_1}{\partial x_2} & \cdots & \frac{\partial \phi_1}{\partial x_m} \\ \frac{\partial \phi_2}{\partial x_1} & \frac{\partial \phi_2}{\partial x_2} & \cdots & \frac{\partial \phi_2}{\partial x_m} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \phi_m}{\partial x_1} & \frac{\partial \phi_m}{\partial x_2} & \cdots & \frac{\partial \phi_m}{\partial x_m} \end{pmatrix} \tag{3}$$

Likewise, γ is expanded about $\hat{x}^-(k)$

$$\gamma(x(k)) = \gamma(\hat{x}^-(k)) + J_\gamma[x(k) - \hat{x}^-(k)] + \cdots \tag{4}$$

where $\hat{x}^-(k)$ is the estimation of the state vector $x(k)$ before measurement at the k-th instant to be receivec and $\hat{x}(k)$ is the updated estimation of the state vector after measurement at the k-th instant has been received. The Jacobian $J_\gamma(x)$ is

$$J_{\gamma}(x) = \frac{\partial \gamma}{\partial x}|_{x=\hat{x}^{-}(k)} = \begin{pmatrix} \frac{\partial \gamma_1}{\partial x_1} & \frac{\partial \gamma_1}{\partial x_2} & \cdots & \frac{\partial \gamma_1}{\partial x_m} \\ \frac{\partial \gamma_2}{\partial x_1} & \frac{\partial \gamma_2}{\partial x_2} & \cdots & \frac{\partial \gamma_2}{\partial x_m} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \gamma_p}{\partial x_1} & \frac{\partial \gamma_p}{\partial x_2} & \cdots & \frac{\partial \gamma_p}{\partial x_m} \end{pmatrix} \tag{5}$$

The resulting expressions create first order approximations of ϕ and γ. Thus the linearized version of the system is obtained:

$$\begin{aligned} x(k+1) &= \phi(\hat{x}(k)) + J_{\phi}(\hat{x}(k))[x(k) - \hat{x}(k)] + w(k) \\ z(k) &= \gamma(\hat{x}^{-}(k)) + J_{\gamma}(\hat{x}^{-}(k))[x(k) - \hat{x}^{-}(k)] + v(k) \end{aligned} \tag{6}$$

Now, the EKF recursion is as follows: First the time update is considered: by $\hat{x}(k)$ the estimation of the state vector at instant k is denoted. Given initial conditions $\hat{x}(0)$ and $P^{-}(0)$ the recursion proceeds as:

- *Measurement update*. Acquire $z(k)$ and compute:

$$\begin{aligned} K(k) &= P^{-}(k)J_{\gamma}^{T}(\hat{x}^{-}(k)) \cdot [J_{\gamma}(\hat{x}^{-}(k))P^{-}(k)J_{\gamma}^{T}(\hat{x}^{-}(k)) + R(k)]^{-1} \\ \hat{x}(k) &= \hat{x}^{-}(k) + K(k)[z(k) - \gamma(\hat{x}^{-}(k))] \\ P(k) &= P^{-}(k) - K(k)J_{\gamma}(\hat{x}^{-}(k))P^{-}(k) \end{aligned} \tag{7}$$

- *Time update*. Compute:

$$\begin{aligned} P^{-}(k+1) &= J_{\phi}(\hat{x}(k))P(k)J_{\phi}^{T}(\hat{x}(k)) + Q(k) \\ \hat{x}^{-}(k+1) &= \phi(\hat{x}(k)) + L(k)u(k) \end{aligned} \tag{8}$$

The schematic diagram of the EKF loop is given in Fig. 1.

2.2 Calculation of local estimations in terms of EIF information contributions

Again the discrete-time nonlinear system of Eq. (1) is considered. The Extended Information Filter (EIF) performs fusion of the local state vector estimates which are provided by the local Extended Kalman Filters, using the *Information matrix* and the *Information state vector* (Lee et al. 2008), (Lee et al. 2008), (Vercauteren & Wang 2005), (Manyika & H. Durrant-Whyte 1994). The Information Matrix is the inverse of the state vector covariance matrix, and can be also associated to the Fisher Information matrix." (Rigatos & Zhang 2009). The Information state vector is the product between the Information matrix and the local state vector estimate

$$\begin{aligned} Y(k) &= P^{-1}(k) = I(K) \\ \hat{y}(k) &= P^{--1}\hat{x}(k) = Y(k)\hat{x}(k) \end{aligned} \tag{9}$$

The update equation for the Information Matrix and the Information state vector are given by

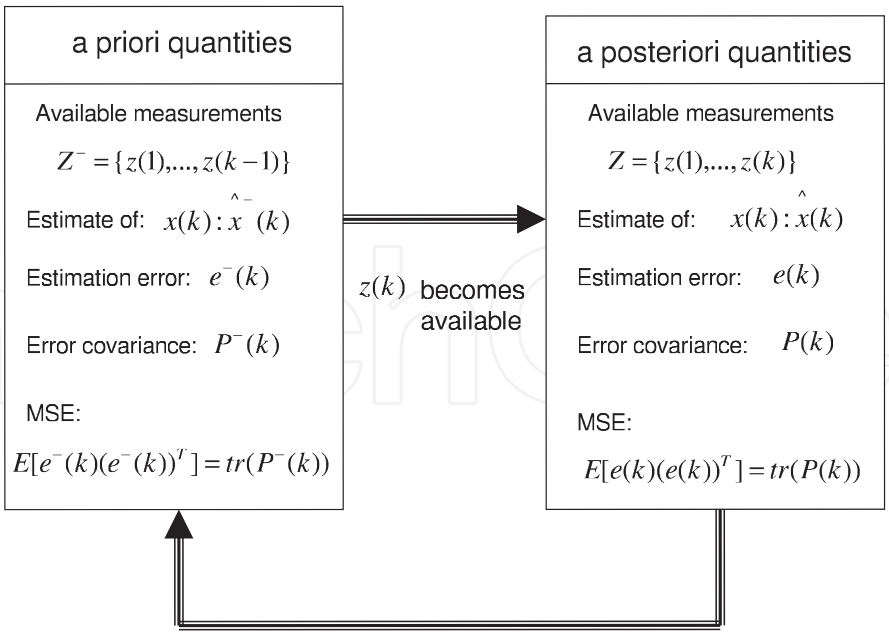

a priori quantities
Available measurements
$Z^{-} = \{z(1),\ldots,z(k-1)\}$
Estimate of: $x(k):\hat{x}^{-}(k)$
Estimation error: $e^{-}(k)$
Error covariance: $P^{-}(k)$
MSE:
$E[e^{-}(k)(e^{-}(k))^{T}] = tr(P^{-}(k))$
$z(k)$ becomes available
a posteriori quantities
Available measurements
$Z = \{z(1),\ldots,z(k)\}$
Estimate of: $x(k):\hat{x}(k)$
Estimation error: $e(k)$
Error covariance: $P(k)$
MSE:
$E[e(k)(e(k))^{T}] = tr(P(k))$

Fig. 1. Schematic diagram of the EKF loop

$$\begin{aligned} Y(k) &= P^{-}(k)^{-1} + J_{\gamma}^{T}(k)R^{-1}(k)J_{\gamma}(k) \\ &= Y^{-}(k) + I(k) \end{aligned} \tag{10}$$

$$\begin{aligned} \hat{y}(k) &= \hat{y}^{-}(k) + J_{\gamma}^{T}R(k)^{-1}[z(k) - \gamma(x(k)) + J_{\gamma}\hat{x}^{-}(k)] \\ &= \hat{y}^{-}(k) + i(k) \end{aligned} \tag{11}$$

where

$$\begin{aligned} &I(k) = J_{\gamma}^{T}(k)R^{(}k) - 1J_{\gamma}(k) \text{ is the associated information matrix and} \\ &i(k) = J_{\gamma}^{T}R^{(}k) - 1[(z(k) - \gamma(x(k))) + J_{\gamma}\hat{x}^{-}(k)] \text{ is the information state contribution} \end{aligned} \tag{12}$$

The predicted information state vector and Information matrix are obtained from

$$\begin{aligned} &\hat{y}^{-}(k) = P^{-}(k)^{-1}\hat{x}^{-}(k) \\ &Y^{-}(k) = P^{-}(k)^{-1} = [J_{\phi}(k)P^{-}(k)J_{\phi}(k)^{T} + Q(k)]^{-1} \end{aligned} \tag{13}$$

The Extended Information Filter is next formulated for the case that multiple local sensor measurements and local estimates are used to increase the accuracy and reliability of the estimation. It is assumed that an observation vector $z^{i}(k)$ is available for N different sensor sites $i = 1, 2, \cdots, N$ and each sensor observes a common state according to the local observation model, expressed by

$$z^{i}(k) = \gamma(x(k)) + v^{i}(k),\ i = 1, 2, \cdots, N \tag{14}$$

where the local noise vector $v^{i}(k) \sim N(0, R^{i})$ is assumed to be white Gaussian and uncorrelated between sensors. The variance of a composite observation noise vector v_{k} is expressed in terms of the block diagonal matrix

$$R(k) = diag[R(k)^1,\ \cdots, R^N(k)]^T \tag{15}$$

The information contribution can be expressed by a linear combination of each local information state contribution i^i and the associated information matrix I^i at the i-th sensor site

$$\begin{gathered} i(k) = \sum_{i=1}^{N} J_\gamma^{i\,T}(k) R^i(k)^{-1}[z^i(k) - \gamma^k(x(k)) + J_\gamma^i(k)\hat{x}^-(k)] \\ I(k) = \sum_{i=1}^{N} J_\gamma^{i\,T}(k) R^i(k)^{-1} J_\gamma^i(k) \end{gathered} \tag{16}$$

Using Eq. (16) the update equations for fusing the local state estimates become

$$\begin{gathered} \hat{y}(k) = \hat{y}^-(k) + \sum_{i=1}^{N} J_\gamma^{i\,T}(k) R^i(k)^{-1}[z^i(k) - \gamma^k(x(k)) + J_\gamma^i(k)\hat{x}^-(k)] \\ Y(k) = Y^-(k) + \sum_{i=1}^{N} J_\gamma^{i\,T}(k) R^i(k)^{-1} J_\gamma^i(k) \end{gathered} \tag{17}$$

It is noted that in the Extended Information Filter an aggregation (master) fusion filter produces a global estimate by using the local sensor information provided by each local filter.

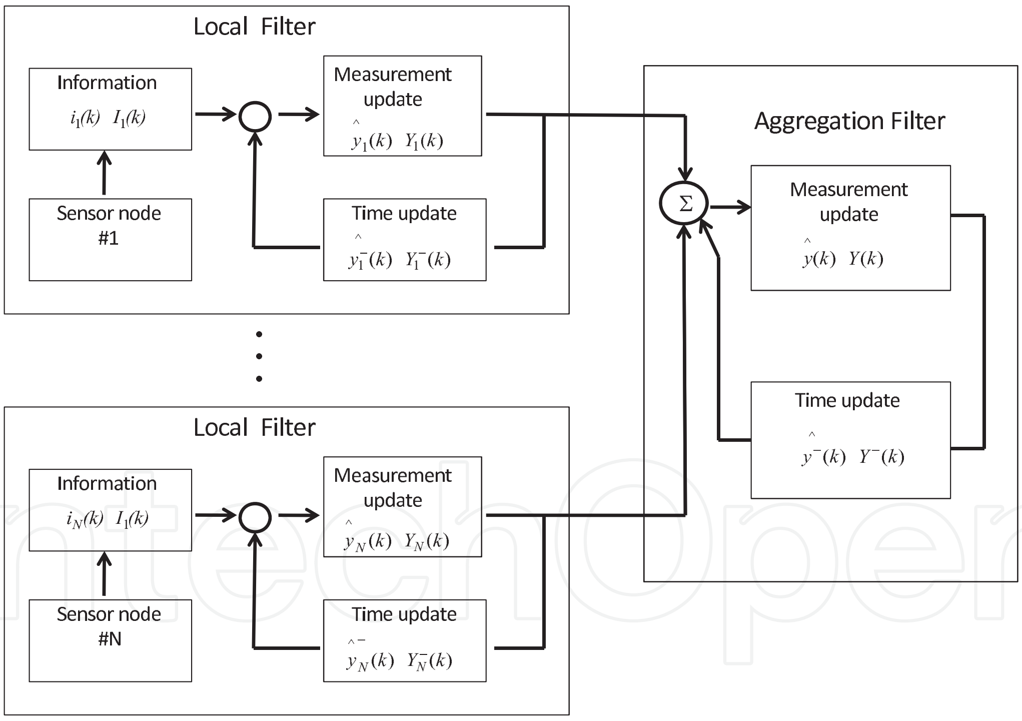


Fig. 2. Fusion of the distributed state estimates with the use of the Extended Information Filter

As in the case of the Extended Kalman Filter the local filters which constitute the Extended Information Filter can be written in terms of *time update* and a *measurement update* equation.
Measurement update: Acquire $z(k)$ and compute

$$\begin{gathered} Y(k) = P^-(k)^{-1} + J_\gamma^T(k) R(k)^{-1} J_\gamma(k) \\ \text{or } Y(k) = Y^-(k) + I(k) \text{ where } I(k) = J_\gamma^T(k) R^{-1}(k) J_\gamma(k) \end{gathered} \tag{18}$$

$$\hat{y}(k) = \hat{y}^-(k) + J_\gamma^T(k)R(k)^{-1}[z(k) - \gamma(\hat{x}(k)) + J_\gamma \hat{x}^-(k)]$$
$$\text{or } \hat{y}(k) = \hat{y}^-(k) + i(k) \tag{19}$$

Time update: Compute

$$Y^-(k+1) = P^-(k+1)^{-1} = [J_\phi(k)P(k)J_\phi(k)^T + Q(k)]^{-1} \tag{20}$$

$$y^-(k+1) = P^-(k+1)^{-1}\hat{x}^-(k+1) \tag{21}$$

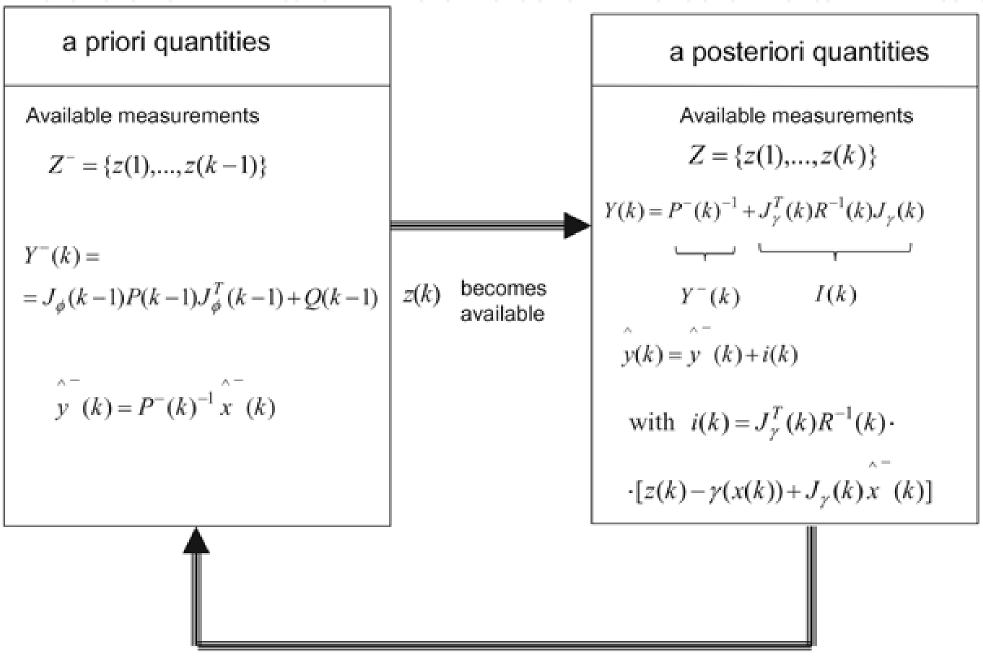

a priori quantities
Available measurements
$Z^- = \{z(1),...,z(k-1)\}$
$Y^-(k) =$
$= J_\phi(k-1)P(k-1)J_\phi^T(k-1) + Q(k-1)$
$\hat{y}^-(k) = P^-(k)^{-1}\hat{x}^-(k)$
$z(k)$ becomes available
a posteriori quantities
Available measurements
$Z = \{z(1),...,z(k)\}$
$Y(k) = \underbrace{P^-(k)^{-1}}_{Y^-(k)} + \underbrace{J_\gamma^T(k)R^{-1}(k)J_\gamma(k)}_{I(k)}$
$\hat{y}(k) = \hat{y}^-(k) + i(k)$
with $i(k) = J_\gamma^T(k)R^{-1}(k)\cdot$
$\cdot[z(k) - \gamma(x(k)) + J_\gamma(k)\hat{x}^-(k)]$

Fig. 3. Schematic diagram of the Extended Information Filter loop

2.3 Extended Information Filtering for state estimates fusion

In the Extended Information Filter each one of the local filters operates independently, processing its own local measurements. It is assumed that there is no sharing of measurements between the local filters and that the aggregation filter (Fig. 2) does not have direct access to the raw measurements feeding each local filter. The outputs of the local filters are treated as measurements which are fed into the aggregation fusion filter (Lee et al. 2008), (Lee et al. 2008), (Vercauteren &Wang 2005). Then each local filter is expressed by its respective error covariance and estimate in terms of information contributions given in Eq.(13)

$$P_i^{-1}(k) = P_i^-(k)^{-1} + J_\gamma^T(k)R(k)^{-1}J_\gamma(k)$$
$$\hat{x}_i(k) = P_i(k)(P_i^-(k)^{-1}\hat{x}_i^-(k) + J_\gamma^T(k)R(k)^{-1}\cdot[z^i(k) - \gamma^k(x(k)) + J_\gamma^i(k)\hat{x}_i^-(k)]) \tag{22}$$

It is noted that the local estimates are suboptimal and also conditionally independent given their own measurements. The global estimate and the associated error covariance for the aggregate fusion filter can be rewritten in terms of the computed estimates and covariances from the local filters using the relations

$$\begin{gathered} J_{\gamma}^{T}(k)R(k)^{-1}J_{\gamma}(k) = P_i^{-1}(k) - P_i^{-}(k)^{-1} \\ J_{\gamma}^{T}(k)R(k)^{-1}[z^i(k) - \gamma^k(x(k)) + J_{\gamma}^{i}(k)\hat{x}_i^{-}(k)] = P_i(k)^{-1}\hat{x}_i(k) - P_i^{-}(k)^{-1}\hat{x}_i^{-}(k) \end{gathered} \tag{23}$$

For the general case of N local filters $i = 1, \cdots, N$, the distributed filtering architecture is described by the following equations

$$\begin{gathered} P(k)^{-1} = P^{-}(k)^{-1} + \sum_{i=1}^{N}[P_i(k)^{-1} - P_i^{-}(k)^{-1}] \\ \hat{x}(k) = P(k)[P^{-}(k)^{-1}\hat{x}^{-}(k) + \sum_{i=1}^{N}(P_i(k)^{-1}\hat{x}_i(k) - P_i^{-}(k)^{-1}\hat{x}_i^{-}(k))] \end{gathered} \tag{24}$$

It is noted that the global state update equation in the above distributed filter can be written in terms of the information state vector and of the information matrix

$$\begin{gathered} \hat{y}(k) = \hat{y}^{-}(k) + \sum_{i=1}^{N}(\hat{y}_i(k) - \hat{y}_i^{-}(k)) \\ \hat{Y}(k) = \hat{Y}^{-}(k) + \sum_{i=1}^{N}(\hat{Y}_i(k) - \hat{Y}_i^{-}(k)) \end{gathered} \tag{25}$$

The local filters provide their own local estimates and repeat the cycle at step $k + 1$. In turn the global filter can predict its global estimate and repeat the cycle at the next time step $k + 1$ when the new state $\hat{x}$ $(k + 1)$ and the new global covariance matrix $P(k + 1)$ are calculated. From Eq. (24) it can be seen that if a local filter (processing station) fails, then the local covariance matrices and the local state estimates provided by the rest of the filters will enable an accurate computation of the system's state vector.

3. Distributed Sigma-Point Kalman Filtering

3.1 Unscented Kalman Filtering at local processing units

It is also possible to estimate the state vectors of the distributed UAVs which constitute the multi-UAV system through the fusion of the estimates provided by local Sigma-Point Kalman Filters. This can be succeeded using the Distributed Sigma-Point Kalman Filter, also known as *Unscented Information Filter* (UIF) (Lee et al. 2008), (Lee et al. 2008). First, the functioning of the local Sigma-Point Kalman Filters will be explained. Each local Sigma-Point Kalman Filter generates an estimation of the UAV's state vector by fusing measurement from distributed sensors (e.g. IMU and GPS). Sigma-Point Kalman Filtering is proposed (Julier et al. 2000), (Julier et al. 2004), (Särrkä 2007). The Sigma-Point Kalman Filter overcomes the flaws of Extended Kalman Filtering. Unlike EKF no analytical Jacobians of the system equations need to be calculated as in the case for the EKF. This makes the sigma-point approach suitable for application in "black-box" models where analytical expressions of the system dynamics are either not available or not in a form which allows for easy linearization. This is achieved through a different approach for calculating the posterior 1st and 2nd order statistics of a random variable that undergoes a nonlinear transformation. The state distribution is represented again by a Gaussian Random Variable but is now specified using a minimal set of deterministically chosen weighted sample points. The basic sigma-point approach can be described as follows:

1. A set of weighted samples (sigma-points) are deterministically calculated using the mean and square-root decomposition of the covariance matrix of the system's state vector. As a minimal requirement the sigma-point set must completely capture the first and second order moments of the prior random variable. Higher order moments can be captured at the cost of using more sigma-points.
2. The sigma-points are propagated through the true nonlinear function using functional evaluations alone, i.e. no analytical derivatives are used, in order to generate a posterior sigma-point set.
3. The posterior statistics are calculated (approximated) using tractable functions of the propagated sigma-points and weights. Typically, these take on the form of a simple weighted sample mean and covariance calculations of the posterior sigma points.

It is noted that the sigma-point approach differs substantially from general stochastic sampling techniques, such as Monte-Carlo integration (e.g Particle Filtering methods) which require significantly more sample points in an attempt to propagate an accurate (possibly non-Gaussian) distribution of the state. The deceptively simple sigma-point approach results in posterior approximations that are accurate to the third order for Gaussian inputs for all nonlinearities. For non-Gaussian inputs, approximations are accurate to at least the second-order, with the accuracy of third and higher-order moments determined by the specific choice of weights and scaling factors.

The Unscented Kalman Filter (UKF) is a special case of Sigma-Point Kalman Filters. The UKF is a discrete time filtering algorithm which uses the unscented transform for computing approximate solutions to the filtering problem of the form

$$\begin{aligned} x(k+1) &= \phi(x(k)) + L(k)U(k) + w(k) \\ y(k) &= \gamma(x(k)) + v(k) \end{aligned} \tag{26}$$

where $x(k) \in R^n$ is the system's state vector, $y(k) \in R^m$ is the measurement, $w(k) \in R^n$ is a Gaussian process noise $w(k) \sim N(0,Q(k))$, and $v(k) \in R^m$ is a Gaussian measurement noise $v(k) \sim N(0,R(k))$. The mean and covariance of the initial state $x(0)$ are $m(0)$ and $P(0)$, respectively.

Some basic operations performed in the UKF algorithm (*Unscented Transform*) are summarized as follows:

1. Denoting the current state mean as $\hat{x}$, a set of $2n+1$ sigma points is taken from the columns of the $n \times n$ matrix $\sqrt{(n+\lambda)P_{xx}}$ as follows:

$$\begin{aligned} x^0 &= \hat{x} \\ x^i &= \hat{x} + [\sqrt{(n+\lambda)}P_{xx}]_i,\ i = 1,\cdots,n \\ x^i &= \hat{x} - [\sqrt{(n+\lambda)}P_{xx}]_i,\ i = n+1,\cdots,2n \end{aligned} \tag{27}$$

and the associate weights are computed:

$$\begin{aligned} W_0^{(m)} &= \frac{\lambda}{(n+\lambda)} \qquad W_0^{(c)} = \frac{\lambda}{(n+\lambda)+(1-\alpha^2+b)} \\ W_i^{(m)} &= \frac{1}{2(n+\lambda)},\ i = 1,\cdots,2n \qquad W_i^{(c)} = \frac{1}{2(n+\lambda)} \end{aligned} \tag{28}$$

where $i = 1, 2, \cdots, 2n$ and $\lambda = \alpha^2(n + \kappa) - n$ is a scaling parameter, while α, β and κ are constant parameters. Matrix P_{xx} is the covariance matrix of the state x.

2. Transform each of the sigma points as

$$z^i = h(x^i) \; i = 0, \cdots, 2n \tag{29}$$

3. Mean and covariance estimates for z can be computed as

$$\begin{gathered} \hat{z} \simeq \sum_{i=0}^{2n} W_i^{(m)} z^i \\ P_{zz} = \sum_{i=0}^{2n} W_i^{(c)} (z^i - \hat{z})(z^i - \hat{z})^T \end{gathered} \tag{30}$$

4. The cross-covariance of x and z is estimated as

$$P_{xz} = \simeq \sum_{i=0}^{2n} W_i^{(c)} (x^i - \hat{x})(z^i - \hat{z})^T \tag{31}$$

The matrix square root of positive definite matrix P_{xx} means a matrix $A = \sqrt{P_{xx}}$ such that $P_{xx} = AA^T$ and a possible way for calculation is SVD.

Next the basic stages of the *Unscented Kalman Filter* are given:

As in the case of the Extended Kalman Filter and the Particle Filter, the Unscented Kalman Filter also consists of prediction stage (time update) and correction stage (measurement update) (Julier et al. 2004), (Särrkä 2007).

Time update: Compute the predicted state mean $\hat{x}^-(k)$ and the predicted covariance $P_{xx}^-(k)$ as

$$\begin{gathered} [\hat{x}^-(k), P_{xx}^-(k)] = UT(f_d, \hat{x}(k-1), P_{xx}(k-1)) \\ P_{xx}^-(k) = P_{xx}(k-1) + Q(k-1) \end{gathered} \tag{32}$$

Measurement update: Obtain the new output measurement z_k and compute the predicted mean $\hat{z}(k)$ and covariance of the measurement $P_{zz}(k)$, and the cross covariance of the state and measurement $P_{xz}(k)$

$$\begin{gathered} [\hat{z}(k), P_{zz}(k), P_{xz}(k)] = UT(h_d, \hat{x}^-(k), P_{xx}^-(k)) \\ P_{zz}(k) = P_{zz}(k) + R(k) \end{gathered} \tag{33}$$

Then compute the filter gain $K(k)$, the state mean $\hat{x}(k)$ and the covariance $P_{xx}(k)$, conditional to the measurement $y(k)$

$$\begin{gathered} K(k) = P_{xz}(k) P_{zz}^{-1}(k) \\ \hat{x}(k) = \hat{x}^-(k) + K(k)[z(k) - \hat{z}(k)] \\ P_{xx}(k) = P_{xx}^-(k) - K(k) P_{zz}(k) K(k)^T \end{gathered} \tag{34}$$

The filter starts from the initial mean $m(0)$ and covariance $P_{xx}(0)$. The stages of state vector estimation with the use of the Unscented Kalman Filter algorithm are depicted in Fig. 6.

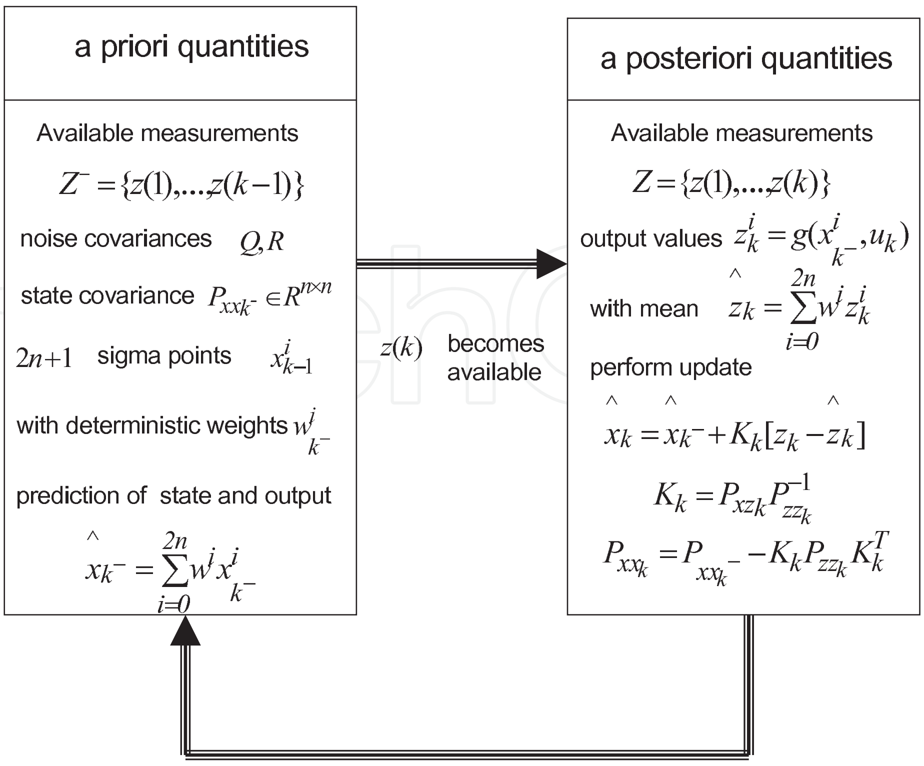

a priori quantities
Available measurements
$Z^- = \{z(1),\dots,z(k-1)\}$
noise covariances Q,R
state covariance $P_{xx_k^-} \in R^{n\times n}$
$2n+1$ sigma points x_{k-1}^i
with deterministic weights $w_{k^-}^j$
prediction of state and output
$\hat{x}_k^- = \sum_{i=0}^{2n} w^i x_{k^-}^i$
$z(k)$ becomes available
a posteriori quantities
Available measurements
$Z = \{z(1),\dots,z(k)\}$
output values $z_k^i = g(x_{k^-}^i, u_k)$
with mean $\hat{z}_k = \sum_{i=0}^{2n} w^i z_k^i$
perform update
$\hat{x}_k = \hat{x}_k^- + K_k[z_k - \hat{z}_k]$
$K_k = P_{xz_k} P_{zz_k}^{-1}$
$P_{xx_k} = P_{xx_k^-} - K_k P_{zz_k} K_k^T$

Fig. 4. Schematic diagram of the Unscented Kalman Filter loop

3.2 Unscented Information Filtering

The Unscented Information Filter (UIF) performs fusion of the state vector estimates which are provided by local Unscented Kalman Filters, by weighting these estimates with local Information matrices (inverse of the local state vector covariance matrices which are again recursively computed) (Lee et al. 2008), (Lee et al. 2008), (Vercauteren &Wang 2005). The Unscented Information Filter is derived by introducing a linear error propagation based on the unscented transformation into the Extended Information Filtering structure. First, an augmented state vector $x_\alpha^-(k)$ is considered, along with the process noise vector, and the associated covariance matrix is introduced

$$\hat{x}_\alpha^-(k) = \begin{pmatrix} \hat{x}^-(k) \\ \hat{w}^-(k) \end{pmatrix}, P^{\alpha -}(k) = \begin{pmatrix} P^-(k) & 0 \\ 0 & Q^-(k) \end{pmatrix} \tag{35}$$

As in the case of local (lumped) Unscented Kalman Filters, a set of weighted sigma points $X_\alpha^{i\,-}(k)$ is generated as

$$\begin{gathered} X_{\alpha,0}^-(k) = \hat{x}_\alpha^-(k) \\ X_{\alpha,i}^-(k) = \hat{x}_\alpha^-(k) + [\sqrt{(n_\alpha + \lambda) P_\alpha^-(k-1)}]_i,\, i = 1,\cdots,n \\ X_{\alpha,i}^-(k) = \hat{x}_\alpha^-(k) + [\sqrt{(n_\alpha + \lambda) P_\alpha^-(k-1)}]_i,\, i = n+1,\cdots,2n \end{gathered} \tag{36}$$

where $\lambda = \alpha^2(n_\alpha + \kappa) - n_\alpha$ is a scaling, while $0 \le \alpha \le 1$ and κ are constant parameters. The corresponding weights for the mean and covariance are defined as in the case of the lumped Unscented Kalman Filter

$$W_0^{(m)} = \frac{\lambda}{n_\alpha + \lambda} \qquad W_0^{(c)} = \frac{\lambda}{(n_\alpha + \lambda) + (1 - \alpha^2 + \beta)}$$
$$W_i^{(m)} = \frac{1}{2(n_\alpha + \lambda)}, i = 1, \cdots, 2n_\alpha \qquad W_i^{(C)} = \frac{1}{2(n_\alpha + \lambda)}, i = 1, \cdots, 2n_\alpha \tag{37}$$

where β is again a constant parameter. The equations of the prediction stage (measurement update) of the information filter, i.e. the calculation of the information matrix and the information state vector of Eq. (13) now become

$$\begin{gathered} \hat{y}^-(k) = Y^-(k)\sum_{i=0}^{2n_\alpha} W_i^m X_i^x(k) \\ Y^-(k) = P^-(k)^{-1} \end{gathered} \tag{38}$$

where X_i^x are the predicted state vectors when using the sigma point vectors X_i^w in the state equation $X_i^x(k+1) = \phi(X_i^{w-}(k)) + L(k)U(k)$. The predicted state covariance matrix is computed as

$$P^-(k) = \sum_{i=0}^{2n_\alpha} W_i^{(c)}[X_i^x(k) - \hat{x}^-(k)][X_i^x(k) - \hat{x}^-(k)]^T \tag{39}$$

As noted, the equations of the Extended Information Filter (EIF) are based on the linearized dynamic model of the system and on the inverse of the covariance matrix of the state vector. However, in the equations of the Unscented Kalman Filter (UKF) there is no linearization of the system dynamics, thus the UKF cannot be included directly into the EIF equations. Instead, it is assumed that the nonlinear measurement equation of the system given in Eq. (1) can be mapped into a linear function of its statistical mean and covariance, which makes possible to use the information update equations of the EIF. Denoting $Y_i(k) = \gamma(X_i^x(k))$ (i.e. the output of the system calculated through the propagation of the i-th sigma point X^i through the system's nonlinear equation) the observation covariance and its cross-covariance are approximated by

$$\begin{aligned} P_{YY}^-(k) &= E[(z(k) - \hat{z}^-(k))(z(k) - \hat{z}^-(k))^T] \\ &\cong J_\gamma(k)P^-(k)J_\gamma(k)^T \end{aligned} \tag{40}$$

$$\begin{aligned} P_{XY}^-(k) &= E[(x(k) - \hat{x}^-(k))(z(k) - \hat{z}^-(k))^T] \\ &\cong P^-(k)J_\gamma(k)^T \end{aligned} \tag{41}$$

where $z(k) = \gamma(x(k))$ and $J_\gamma(k)$ is the Jacobian of the output equation $\gamma(x(k))$. Next, multiplying the predicted covariance and its inverse term on the right side of the information matrix Eq. (12) and replacing $P(k)J_\gamma(k)^T$ with $P_{XY}^-(k)$ (k) gives the following representation of the information matrix (Lee et al. 2008), (Lee et al. 2008), (Vercauteren &Wang 2005)

$$\begin{aligned} I(k) &= J_\gamma(k)^T R(k)^{-1} J_\gamma(k) \\ &= P^-(k)^{-1} P^-(k) J_\gamma(k)^T R(k)^{-1} J_\gamma^-(k) P^-(k)^T (P^-(k)^{-1})^T \\ &= P^-(k)^{-1} P_{XY}(k) R(k)^{-1} P_{XY}(k)^T (P^-(k)^{-1})^T \end{aligned} \tag{42}$$

where $P^{-}(k)^{-1}$ is calculated according to Eq. (39) and the cross-correlation matrix $P_{XY}(k)$ is calculated from

$$P_{XY}^{-}(k)=\sum_{i=0}^{2n_{\alpha}}W_i^{(c)}[X_i^x(k)-\hat{x}^-(k)][Y_i(k)-\hat{z}^-(k)]^T \tag{43}$$

where $Y_i(k)=\gamma(X_i^x(k))$ and the predicted measurement vector $\hat{z}^-(k)$ is obtained by $\hat{z}^-(k)=\sum_{i=0}^{2n}W_i^{(m)}Y_i(k)$. Similarly, the information state vector i_k can be rewritten as

$$\begin{aligned}
i(k)&=J_\gamma^T(k)R(k)^{-1}[z(k)-\gamma(x(k))+J_\gamma^T(k)\hat{x}^-(k)]\\
&=P^-(k)^{-1}P^-(k)J_\gamma^T(k)R(k)^{-1}\cdot\\
&\cdot[z(k)-\gamma(x(k))+J_\gamma^T(k)\hat{x}^-(k)(P^-(k))^T(P^-(k)^{-1})^T\hat{x}^-(k)]\\
&=P^-(k)^{-1}P_{XY}^-(k)R(k)^{-1}[z(k)-\gamma(x(k))+P_{XY}^-(k)(P^-(k)^{-1})^T\hat{x}^-(k)]
\end{aligned} \tag{44}$$

To complete the analogy to the information contribution equations of the EIF a ″measurement″ matrix $H^T(k)$ is defined as

$$H(k)^T=P^-(k)^{-1}P_{XY}^-(k) \tag{45}$$

In terms of the measurement matrix $H(k)$ the information contributions equations are written as

$$\begin{aligned}
i(k)&=H^T(k)R(k)^{-1}[z(k)-\gamma(x(k))+H(k)\hat{x}^-(k)]\\
I(k)&=H^T(k)R(k)^{-1}H(k)
\end{aligned} \tag{46}$$

The above procedure leads to an implicit linearization in which the nonlinear measurement equation of the system given in Eq. (1) is approximated by the statistical error variance and its mean

$$z(k)=h(x(k))\simeq H(k)x(k)+\overline{u}(k) \tag{47}$$

where $\overline{u}(k)=\gamma(\hat{x}^-(k))-H(k)\hat{x}^-(k)$ is a measurement residual term. (47).

3.3 Calculation of local estimations in terms of UIF information contributions

Next, the local estimations provided by distributed (local) Unscented Kalmans filters will be expressed in terms of the information contributions (information matrix I and information state vector i) of the Unscented Information Filter, which were defined in Eq. (46) (Lee et al. 2008), (Lee et al. 2008), (Vercauteren &Wang 2005). It is assumed that the observation vector $\overline{z}_i(k+1)$ is available from N different sensors, and that each sensor observes a common state according to the local observation model, expressed by

$$\overline{z}_i(k)=H_i(k)x(k)+\overline{u}_i(k)+v_i(k) \tag{48}$$

where the noise vector $v_i(k)$ is taken to be white Gaussian and uncorrelated between sensors. The variance of the composite observation noise vector v_k of all sensors is written in terms of the block diagonal matrix $R(k)$ = $diag[R_1(k)^T, \cdots, R_N(k)^T]^T$. Then one can define the local information matrix $I_i(k)$ and the local information state vector $i^i(k)$ at the i-th sensor, as follows

$$\begin{gathered} i_i(k) = H_i^T(k)R_i(k)^{-1}[z_i(k) - \gamma_i(x(k)) + H_i(k)\hat{x}^-(k)] \\ I_i(k) = H_i^T(k)R_i(k)^{-1}H_i(k) \end{gathered} \tag{49}$$

Since the information contribution terms have group diagonal structure in terms of the innovation and measurement matrix, the update equations for the multiple state estimation and data fusion are written as a linear combination of the local information contribution terms

$$\begin{gathered} \hat{y}(k) = \hat{y}^-(k) + \sum_{i=1}^{N} i_i(k) \\ Y(k) = Y^-(k) + \sum_{i=1}^{N} I_i(k) \end{gathered} \tag{50}$$

Then using Eq. (38) one can find the mean state vector for the multiple sensor estimation problem.

As in the case of the Unscented Kalman Filter, the Unscented Information Filter running at the *i*-th measurement processing unit can be written in terms of *measurement update* and *time update* equations:

Measurement update: Acquire measurement $z(k)$ and compute

$$\begin{gathered} Y(k) = P^-(k)^{-1} + H^T(k)R^{-1}(k)H(k) \\ \text{or } Y(k) = Y^-(k) + I(k) \text{ where } I(k) = H^T(k)R^{-1}(k)H(k) \end{gathered} \tag{51}$$

$$\begin{gathered} \hat{y}(k) = \hat{y}^-(k) + H^T(k)R^{-1}(k)[z(k) - \gamma(\hat{x}(k)) + H(k)\hat{x}^-(k)] \\ \text{or } \hat{y}(k) = \hat{y}^-(k) + i(k) \end{gathered} \tag{52}$$

Time update: Compute

$$\begin{gathered} Y^-(k+1) = (P^-(k+1))^{-1} \\ \text{where } P^-(k+1) = \sum_{i=0}^{2n_\alpha} W_i^{(c)}[X_i^x(k+1) - \hat{x}^-(k+1)][X_i^x(k+1) - \hat{x}^-(k+1)]^T \end{gathered} \tag{53}$$

$$\begin{gathered} \hat{y}(k+1) = Y(k+1)\sum_{i=0}^{2n_\alpha} W_i^{(m)} X_i^x(k+1) \\ \text{where } X_i^x(k+1) = \phi(X_i^w(k)) + L(k)U(k) \end{gathered} \tag{54}$$

3.4 Distributed Unscented Information Filtering for state estimates fusion

It has been shown that the update of the aggregate state vector of the Unscented Information Filter architecture can be expressed in terms of the local information matrices I_i and of the local information state vectors i_i, which in turn depend on the local covariance matrices P and cross-covariance matrices P_{XY}. Next, it will be shown that the update of the aggregate state vector can be also expressed in terms of the local state vectors $x_i(k)$ and in terms of the local covariance matrices $P_i(k)$ and cross-covariance matrices $P_{XY}^i(k)$. It is assumed that the local filters do not have access to each other row measurements and that they are allowed to communicate only their information matrices and their local information state vectors. Thus each local filter is expressed by its respective error covariance and estimate in terms of the local information state contribution i_i and its associated information matrix I_i at the *i*-th filter site. Then using Eq. (38) one obtains

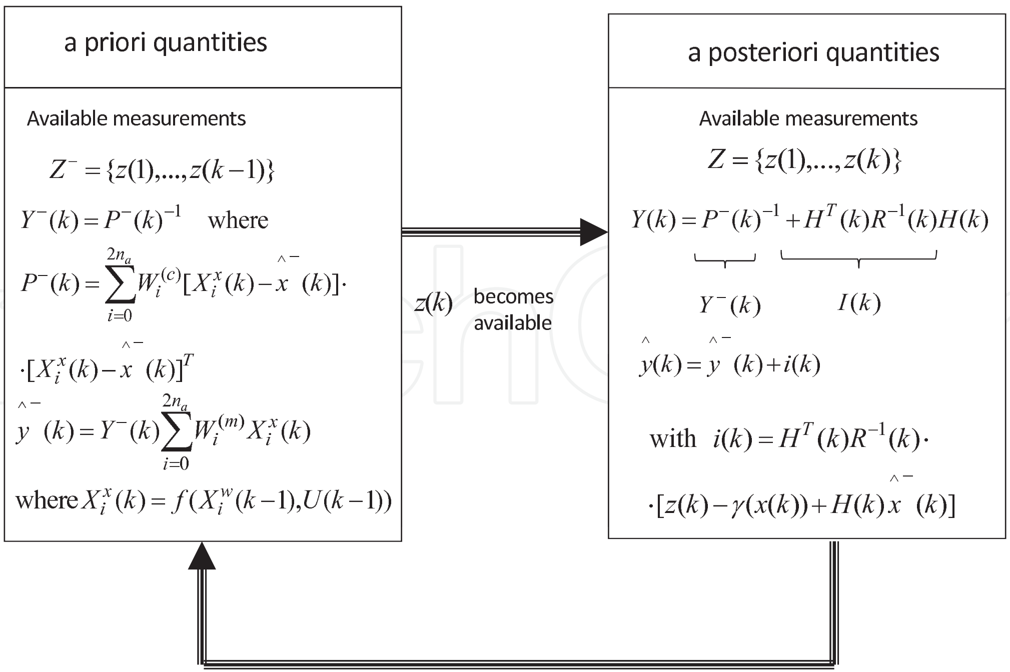

a priori quantities
Available measurements
$Z^- = \{z(1),\ldots,z(k-1)\}$
$Y^-(k) = P^-(k)^{-1}$ where
$P^-(k) = \sum_{i=0}^{2n_a} W_i^{(c)}[X_i^x(k) - \overset{\wedge -}{x}(k)]\cdot$
$\cdot[X_i^x(k) - \overset{\wedge -}{x}(k)]^T$
$\overset{\wedge -}{y}(k) = Y^-(k)\sum_{i=0}^{2n_a} W_i^{(m)} X_i^x(k)$
where $X_i^x(k) = f(X_i^w(k-1), U(k-1))$
$z(k)$ becomes available
a posteriori quantities
Available measurements
$Z = \{z(1),\ldots,z(k)\}$
$Y(k) = \underbrace{P^-(k)^{-1}}_{Y^-(k)} + \underbrace{H^T(k)R^{-1}(k)H(k)}_{I(k)}$
$\hat{y}(k) = \overset{\wedge -}{y}(k) + i(k)$
with $i(k) = H^T(k)R^{-1}(k)\cdot$
$\cdot[z(k) - \gamma(x(k)) + H(k)\overset{\wedge -}{x}(k)]$

Fig. 5. Schematic diagram of the Unscented Information Filter loop

$$\begin{aligned} P_i(k)^{-1} &= P_i^-(k)^{-1} + H_i^T(k)R_i(k)^{-1}H_i(k) \\ \hat{x}_i &= P_i(k)(P_i^-(k)\hat{x}_i^-(k) + H_i^T(k)R_i(k)^{-1}[z_i(k) - \gamma_i(x(k)) + H_i(k)\hat{x}^-(k)]) \end{aligned} \tag{55}$$

Using Eq. (55), each local information state contribution i_i and its associated information matrix I_i at the i-th filter are rewritten in terms of the computed estimates and covariances of the local filters

$$\begin{aligned} H_i^T(k)R_i(k)^{-1}H_i(k) &= P_i(k)^{-1} - P_{\,i}^-(k)^{-1} \\ H_i^T(k)R_i(k)^{-1}[z_i(k) - \gamma_i(x(k)) + H_i(k)\hat{x}^-(k)] &= P_i(k)^{-1}\hat{x}_i(k) - P_i^-(k)^{-1}\hat{x}_i^-(k) \end{aligned} \tag{56}$$

where according to Eq.(45) it holds $H_i(k) = P_i^-(k)^{-1}P_{XY,i}^-(k)$. Next, the aggregate estimates of the distributed unscented information filtering are derived for a number of N local filters $i = 1, \cdots, N$ and sensor measurements, first in terms of covariances (Vercauteren &Wang 2005), (Lee et al. 2008), (Lee et al. 2008)

$$\begin{aligned} P(k)^{-1} &= P^-(k)^{-1} + \sum_{i=1}^{N}[P_i(k)^{-1} - P_i^-(k)^{-1}] \\ \hat{x}(k) &= P(k)[P^-(k)^{-1}\hat{x}^-(k) + \sum_{i=1}^{N}(P_i(k)^{-1}\hat{x}_i(k) - P_i^-(k)^{-1}\hat{x}_i^-(k))] \end{aligned} \tag{57}$$

and also in terms of the information state vector and of the information state covariance matrix

$$\begin{aligned} \hat{y}(k) &= \hat{y}^-(k) + \sum_{i=1}^{N}(\hat{y}_i(k) - \hat{y}_i^-(k)) \\ Y(k) &= Y^-(k) + \sum_{i=1}^{N}[Y_i(k) - Y_i^-(k)] \end{aligned} \tag{58}$$

State estimation fusion based on the Unscented Information Filter (UIF) is fault tolerant. From Eq. (57) it can be seen that if a local filter (processing station) fails, then the local covariance matrices and local estimates provided by the rest of the filters will enable a reliable calculation of the system's state vector. Moreover, it is and computationally efficient comparing to centralized filters and results in enhanced estimation accuracy.

4. Distributed Particle Filter

4.1 Particle Filtering at local processing units

4.1.1 The particle approximation of probability density functions

One can also estimate the state vector of the UAVs that constitute the multi-UAV system through the fusion of estimates provided by local Particle Filters. This can be succeeded using the Distributed Particle Filter (DPF). First, the functioning of the local Particle Filters will be explained. Each local Particle Filter generates an estimation of the UAV's state vector by fusing measurements from distibuted sensors. Particle Filtering is a method for state estimation that is not dependent on the probability density function of the measurements. In the general case the equations of the optimal filter used for the calculation of the state-vector of a dynamical system do not have an explicit solution. This happens for instance when the process noise and the noise of the output measurement do not follow a Gaussian distribution. In that case approximation through Monte-Carlo methods can be used (Thrun wt al. 2005). A sampling of size N is assumed, i.e. N i.i.d. (independent identically distributed) variables $\xi^1, \xi^2, \cdots, \xi^N$. This sampling follows the p.d.f. $p(x)$ i.e. $\xi^{1:N} \sim p(x)$. Instead of $p(x)$ the function $p(x) \simeq p^N(x) = \frac{1}{N}\sum_{i=1}^{N}\delta_{\xi^i}(x)$ can be used. It is assumed that all points ξ^i have an equal weighted contribution to the approximation of $p(x)$. A more general approach would be if weight factors were assigned to the points ξ^i, which will also satisfy the normality condition $\sum_{i=1}^{N} w^i = 1$. In the latter case

$$p(x) \simeq p^N(x) = \sum_{i=1}^{N} w^i \delta_{\xi^i}(x) \tag{59}$$

If $p(\xi^i)$ is known then the probability $P(x)$ can be approximated using the discrete values of the p.d.f. $p(\xi^i) = w^i$. If sampling over the p.d.f. $p(x)$ is unavailable, then one can use a p.d.f. $\bar{p}(x)$ with similar support set, i.e. $p(x) = 0 \Rightarrow \bar{p}(x) = 0$. Then it holds $E(\phi(x)) = \int \phi(x)p(x)dx = \int \phi(x)\bar{p}(x)\frac{p(x)}{\bar{p}(x)}dx$. If the N samples of $\bar{p}(x)$ are available at the points $\tilde{\xi}^1 \cdots \tilde{\xi}^N$, i.e. $\bar{p}(\tilde{\xi})^i = \delta_{\tilde{\xi}^i}(x)$ and the weight coefficients w^i are defined as $w^i = \frac{p(\tilde{\xi}^i)}{\bar{p}(\tilde{\xi}^i)}$, then it is easily shown that

$$E(\phi(x)) \simeq \sum_{i=1}^{N} w^i \phi(\tilde{\xi}^i), \quad \text{where} \begin{cases} \tilde{\xi}^{1:N} \sim \bar{p}(x) \\ w^i = p(\tilde{x}^i) / \bar{p}(\tilde{x}^i) \end{cases} \tag{60}$$

The meaning of Eq. (60) is as follows: assume that the p.d.f. $p(x)$ is unknown (target distribution), however the p.d.f. $\bar{p}(x)$ (importance law) is available. Then, it is sufficient to sample on $\bar{p}(x)$ and find the associated weight coefficients w^i so as to calculate $E(\phi(x))$.

4.1.2 The prediction stage

As in the case of the Kalman Filter or the Extended Kalman Filter the particles filter consists of the measurement update (correction stage) and the time update (prediction stage) (Rigatos 2009b),(Thrun wt al. 2005). The prediction stage calculates $p(x(k)\,|\,Z^{-})$ where $Z^{-} = \{z(1), z(2), \cdots, z(n-1)\}$ according to Eq. (59). It holds that:

$$p(x(k-1)\,|\,Z^{-}) = \sum_{i=1}^{N} w_{k-1}^{i}\delta_{\xi_{k-1}^{i}}(x(k-1)) \tag{61}$$

while from Bayes formula it holds $p(x(k)\,|\,Z^{-}) = \int p(x(k)\,|\,x(k-1))p(x(k-1)\,|\,Z^{-})dx$. Using also Eq. (61) one finally obtains

$$\begin{gathered} p(x(k)\,|\,Z^{-}) = \sum_{i=1}^{N} w_{k-1}^{i}\delta_{\xi_{k^-}^{i}}(x(k)) \\ \text{with } \xi_{k^-}^{i} \sim p(x(k)\,|\,x(k-1) = \xi_{k-1}^{i}) \end{gathered} \tag{62}$$

The meaning of Eq. (62) is as follows: the state equation of the system is executed N times, starting from the N previous values of the state vectors $x(k-1) = \xi_{k-1}^{i}$

$$\begin{gathered} \hat{x}(k+1) = \phi(\hat{x}(k)) + L(k)u(k) + w(k) \\ z(k) = \gamma(\hat{x}(k)) + v(k) \end{gathered} \tag{63}$$

Thus estimations of the current value of the state vector $\hat{x}(k)$ are obtained, and consequently the mean value of the state vector will be given from Eq. (62). This means that the value of the state vector which is calculated in the prediction stage is the result of the weighted averaging of the state vectors which were calculated after running the state equation, starting from the N previous values of the state vectors ξ_{k-1}^{i}.

4.1.3 The correction stage

The a-posteriori probability density is found using Eq. (62). Now a new position measurement $z(k)$ is obtained and the objective is to calculate the corrected probability density $p(x(k)\,|\,Z)$, where $Z = \{z(1), z(2), \cdots, z(k)\}$. From Bayes law it holds that $p(x(k)\,|\,Z) = \frac{p(Z\,|\,x(k))p(x(k))}{p(Z)}$ which can be also written as

$$p(x(k)\,|\,Z) = \frac{p(z(k)\,|\,x(k))p(x(k)\,|\,Z^{-})}{\int p(z(k)\,|\,x(k),Z^{-})p(x(k)\,|\,Z^{-})dx} \tag{64}$$

Substituting Eq. (62) into Eq. (64) and after intermediate calculations one finally obtains

$$\begin{gathered} p(x(k)\,|\,Z) = \sum_{i=1}^{N} w_{k}^{i}\delta_{\xi_{k^-}^{i}}(x(k)) \\ \text{where} \quad w_{k}^{i} = \frac{w_{k^-}^{i}p(z(k)\,|\,x(k) = \xi_{k^-}^{i})}{\sum_{j=1}^{N} w_{k^-}^{j}p(z(k)\,|\,x(k) = \xi_{k^-}^{j})} \end{gathered} \tag{65}$$

Eq. (65) denotes the corrected value for the state vector. The recursion of the Particle Filter proceeds in a way similar to the update of the Kalman Filter or the Extended Kalman Filter, i.e.:

- *Measurement update*: Acquire $z(k)$ and compute

$$\begin{gathered}\text{new value of the state vector}\\ p(x(k)\,|\,Z)=\sum_{i=1}^{N}w_k^i\delta_{\xi^i_{k^-}}(x(k))\\ \text{with corrected weights}\\ w_k^i=\frac{w^i_{k^-}p(z(k)\,|\,x(k)=\xi^i_{k^-})}{\sum_{j=1}^{N}w^i_{k^-}p(z(k)\,|\,x(k)=\xi_{k^-})^i}\text{and } \xi_k^i=\xi^i_{k^-}\end{gathered}\tag{66}$$

Resampling for substitution of the degenerated particles

- *Time update*: compute state vector $x(k+1)$ according to the pdf

$$\begin{gathered}p(x(k+1)\,|\,Z)=\sum_{i=1}^{N}w_k^i\delta_{\xi_k^i}(x(k))\\ \text{where } \xi_k^i\sim p(x(k+1)\,|\,x(k)=\xi_k^i)\end{gathered}\tag{67}$$

The stages of state vector estimation with the use of the Particle Filtering algorithm are depicted in Fig. 6.

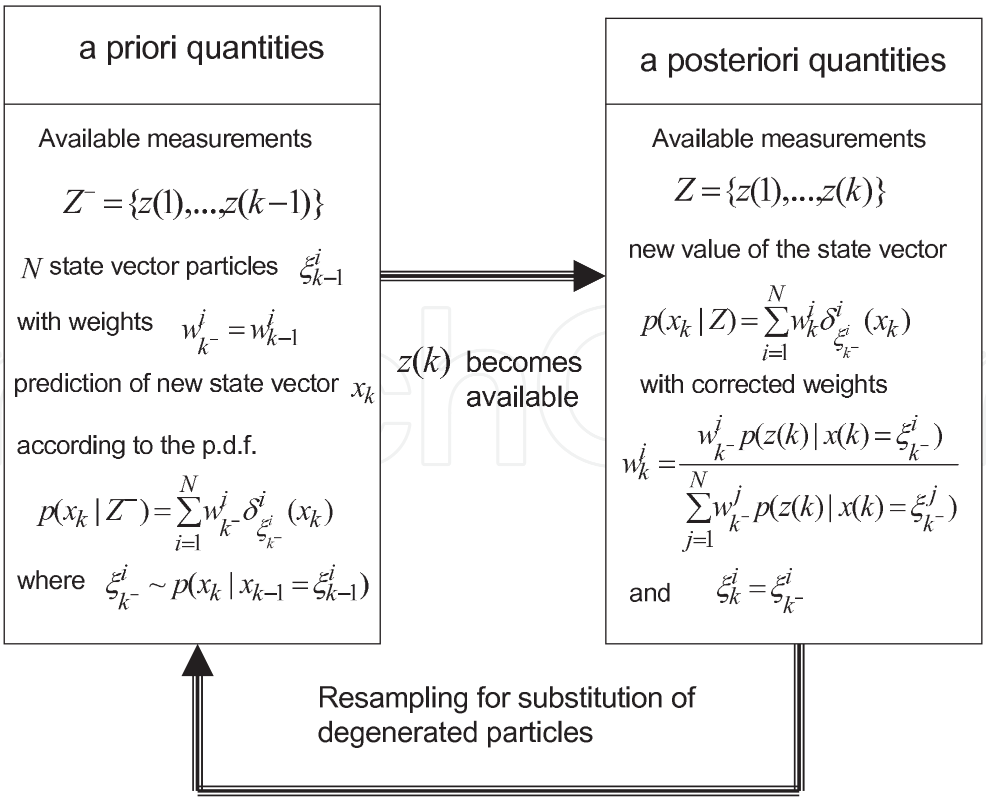

a priori quantities
Available measurements
$Z^-=\{z(1),\ldots,z(k-1)\}$
N state vector particles ξ^i_{k-1}
with weights $w^j_{k^-}=w^j_{k-1}$
prediction of new state vector x_k
according to the p.d.f.
$p(x_k\,|\,Z^-)=\sum_{i=1}^{N}w^j_{k^-}\delta^i_{\xi^i_{k^-}}(x_k)$
where $\xi^i_{k^-}\sim p(x_k\,|\,x_{k-1}=\xi^i_{k-1})$
$z(k)$ becomes available
a posteriori quantities
Available measurements
$Z=\{z(1),\ldots,z(k)\}$
new value of the state vector
$p(x_k\,|\,Z)=\sum_{i=1}^{N}w^j_k\delta^i_{\xi^i_{k^-}}(x_k)$
with corrected weights
$w^j_k=\frac{w^j_{k^-}p(z(k)\,|\,x(k)=\xi^i_{k^-})}{\sum_{j=1}^{N}w^j_{k^-}p(z(k)\,|\,x(k)=\xi^j_{k^-})}$
and $\xi^i_k=\xi^i_{k^-}$
Resampling for substitution of degenerated particles

Fig. 6. Schematic diagram of the Particle Filter loop

4.1.4 Resampling issues in particle filtering

The algorithm of particle filtering which is described through Eq. (62) and Eq. (65) has a significant drawback: after a certain number of iterations *k*, almost all the weights w_k^i become 0. In the ideal case all the weights should converge to the value $\frac{1}{N}$, i.e. the particles should have the same significance. The criterion used to define a sufficient number of particles is $N_k^{\text{eff}} = \frac{1}{\sum_{i=1}^{N} w_k^{i\,2}} \in [1,N]$. When N_k^{eff} is close to value *N* then all particles have almost the same significance. However using the algorithm of Eq. (62) and Eq. (65) results in $N_k^{\text{eff}} \to 1$, which means that the particles are degenerated, i.e. they lose their effectiveness. Therefore, it is necessary to modify the algorithm so as to assure that degeneration of the particles will not take place (Rigatos 2009a), (Thrun wt al. 2005), (Zhang et al. 2005).

When N_k^{eff} is small then most of the particles have weights close to 0 and consequently they have a negligible contribution to the estimation of the state vector. To overcome this drawback of the PF algorithm weakens such particles in favor of particles that have a non-negligible contribution. Therefore, the particles of low weight factors are removed and their place is occupied by duplicates of the particles with high weight factors. The total number of particles remains unchanged (equal to *N*) and therefore this procedure can be viewed as a ″resampling″ or ″redistribution″ of the particles set.

The particles resampling presented above maybe slow if not appropriately tuned. There are improved versions of it which substitute the particles of low importance with those of higher importance. A first choice would be to perform a multinomial resampling. *N* particles are chosen between $\{\xi_k^1,\cdots,\xi_k^N\}$ and the corresponding weights are $w_k^1,\cdots,w_k^N$. The number of times each particle is selected is given by $[j_1, \cdots, j_n]$. Thus a set of *N* particles is again created, the elements of which are chosen after sampling with the discrete distribution $\sum_{i=1}^{N} w_k^i \delta_{\xi_k^i}(x)$. The particles $\{\xi_k^1,\cdots,\xi_k^N\}$ are chosen according to the probabilities $\{w_k^1,\cdots,w_k^N$ }. The selected particles are assigned with equal weights $\frac{1}{N}$.

Although sorting of the particles' weights is not necessary for the convergence of the particle filter algorithm, there are variants of the resampling procedure of ($\xi_k^i, w_k^i \; i = 1,\cdots,N$) which are based on previous sorting in decreasing order of the particles' weights (efficient sorting approaches make the complexity of the particle filtering to be $O(Nlog(N))$, while the avoidance of resampling results in a faster algorithm of complexity $O(N)$). Sorting of particles' weights gives $w^{s[1]} > w^{s[2]} > \cdots > w^{s[N]}$. A random numbers generator is evoked and the resulting numbers $u^{i:N} \sim U[0,1]$ fall in the partitions of the interval [0,1]. The width of these partitions is w^i and thus a redistribution of the particles is generated. For instance, in a wide partition of width w^j will be assigned more particles than to a narrow partition of witdh w^m. A detailed analysis on the tuning of the resampling procedure in Particle Fitlering has been given in (Rigatos 2009a).

4.2 Distributed Particle Filtering for state estimation fusion

The Distributed Particle Filter performs fusion of the state vector estimates which are provided by the local Particle Filters. This is succeeded by fusing the discrete probability density functions of the local Particle Filters into a common probability distribution of the system’s state vector. Without loss of generality fusion between two estimates which are provided by two different probabilistic estimators (particle filters) is assumed. This amounts to a multiplication and a division operation to remove the common information, and is given by (Ong et al. 2008), (Ong et al. 2006)

$$p(x(k)\,|\,Z_A \bigcup Z_B) \propto \frac{p(x(k)\,|\,Z_A)p(x(k)\,|\,Z_B)}{p(x(k)\,|\,Z_A \bigcap Z_B)} \tag{68}$$

where Z_A is the sequence of measurements associated with the *i*-th processing unit and Z_B is the sequence of measurements associated with the *j*-th measurement unit. In the implementation of distributed particle filtering, the following issues arise:

1. Particles from one particle set (which correspond to a local particle filter) do not have the same support (do not cover the same area and points on the samples space) as particles from another particle set (which are associated with another particle filter). Therefore a point-to-point application of Eq. (68) is not possible.
2. The communication of particles representation (i.e. local particle sets and associated weight sets) requires significantly more bandwidth compared to other representations, such as Gaussian mixtures.

Fusion of the estimates provided by the local particle filters (located at different processing units) can be performed through the following stages. First, the discrete particle set of Particle Filter *A* (Particle Filter *B*) is transformed into a continuous distribution by placing a Gaussian kernel over each sample (Fig. 7) (Musso et al. 2001)

$$K_h(x) = h^2 K(x) \tag{69}$$

where $K()$ is the rescaled Kernel density and $h > 0$ is the scaling parameter. Then the continuous distribution *A* (*B*) is sampled with the other particles set *B* (*A*) to obtain the new importance weights, so that the weighted sample corresponds to the numerator of Eq. (68) (Fig. 8). Such a conversion from a discrete particle probability distribution functions $\sum_{i=1}^{N} w_A^{(i)}\delta(x_A^{(i)})$ $(\sum_{i=1}^{N} w_B^{(i)}\delta(x_B^{(i)}))$ into continuous distributions is denoted as

$$\sum_{i=1}^{N} w_A^{(i)}\delta(x_A^{(i)}) \to p_A(x) \ \ (\sum_{i=1}^{N} w_B^{(i)}\delta(x_B^{(i)}) \to p_B(x)) \tag{70}$$

The common information appearing in the processing units *A* and *B* should not be taken into account in the joint probability distribution which is created after fusing the local probability densities of *A* and *B*. This means that in the joint p.d.f. one should sample with importance weights calculated according to Eq. (68). The objective is then to create an importance sampling approximation for the joint distribution that will be in accordance to Eq. (68). A solution to this can be obtained through Monte Carlo sampling and suitable selection of the so-called ″proposal distribution″ (Ong et al. 2008), (Ong et al. 2006)]

According to the above, for the joint distribution the idea behind Monte Carlo sampling is to draw *N* i.i.d samples from the associated probability density function $p(x)$, such that the target density is approximated by a point-mass function of the form

$$p(x) \simeq \sum_{i=1}^{N} w_k^{(i)}\delta(x_k^{(i)}) \tag{71}$$

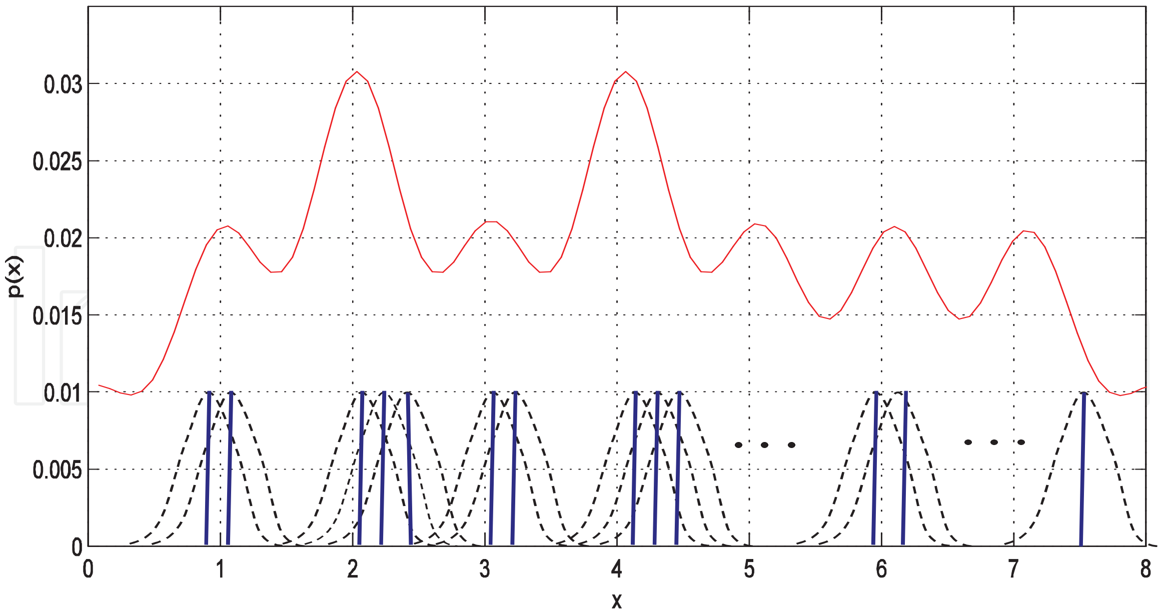


Fig. 7. Conversion of the particles discrete probability density function to a continuous distribution, after allocating a Gaussian kernel over each particle

where $\delta(x_k^{(i)})$ is a Dirac delta mass located at $x_k^{(i)}$. Then the expectation of some function $f(x)$ with respect to the pdf $p(x)$ is given by

$$I(f) = E_{p(x)}[f(x)] = \int f(x)p(x)dx \tag{72}$$

the Monte-Carlo approximation of the integral with samples is then

$$I_N(f) = \frac{1}{N}\sum_{i=1}^{N} f(x^{(i)}) \tag{73}$$

where $x^{(i)} \simeq p(X)$ and $I_N(f) \rightarrow I(f)$ for $N \rightarrow \infty$. since, the true probability distribution $p(x)$ is hard to sample from, the concept of importance sampling is to select a proposal distribution $\overline{p}(x)$ in place of $p(x)$, with the assumption that $\overline{p}(x)$ includes the support space of $p(x)$. Then the expectation of function $f(x)$, previously given in Eq. (72), is now calculated as

$$I(f) = \int f(x)\frac{p(x)}{\overline{p}(x)}\overline{p}(x)dx = \int f(x)w(x)\overline{p}(x)dx \tag{74}$$

where $w(x)$ are the importance weights

$$w(x) = \frac{p(x)}{\overline{p}(x)} \tag{75}$$

Then the Monte-Carlo estimation of the mean value of function $f(x)$ becomes

$$I_N(f) = \sum_{i=1}^{N} f(x^{(i)})w(x^{(i)}) \tag{76}$$

For the division operation, the desired probability distribution is

$$p(x^{(i)}) = \frac{p_A(x^{(i)})}{p_B(x^{(i)})} \tag{77}$$

In that case the important weights of the fused probability density functions become

$$w(x^{(i)}) = \frac{p_A(x^{(i)})}{p_B(x^{(i)})\overline{p}(x^{(i)})} \tag{78}$$

which is then normalized so that $\sum_{i=1}^{N} w(x^{(i)}) = 1/N$, where N is the number of particles. The next step is to decide what will be the form of the proposal distribution $\overline{p}(x)$. A first option is to take $\overline{p}(x)$ to be a uniform distribution, with a support that covers both of the support sets of the distributions A and B.

$$\overline{p}(x) = U(x) \tag{79}$$

Then the sample weights $\overline{p}(x^{(i)})$ are all equal at a constant of value C. Hence the importance weights are

$$w(x^{(i)}) = \frac{p_A(x^{(i)})}{p_B(x^{(i)})C} \tag{80}$$

Another suitable proposal distribution that takes more into account the new information re ceived (described as the probability distribution of the second processing unit) is given by

$$\overline{p}(x) = p_B(x) \tag{81}$$

and the important weights are then adjusted to be

$$w(x^{(i)}) = \frac{p_A(x^{(i)})}{p_B(x^{(i)})^2} \tag{82}$$

5. Nonlinear control for autonomous UAV navigation

5.1 Kinematic model of the UAV

For the design of the autonomous navigation system of the UAVs a suitable control scheme has to be chosen. In this control loop there will be processing of the estimated UAV state vector, as obtained through the distributed filtering algorithms which were presented in Sections 2 to 4. To this end, the kinematic model the kinematic model of the UAVs has to be analyzed first. Based on this kinematic model a flatness-based controller will be derived. The UAV dynamics suggest the following structure for constant altitude manoeuvres (Léchevin & Rabbath 2006):

$$\begin{gathered} \dot{x} = vcos(\theta),\ \dot{y} = vsin(\theta),\ \dot{\theta} = u_1 \\ \dot{v} = u_2,\ \dot{h} = 0 \end{gathered} \tag{83}$$

where (x,y) is the desired inertial position of the UAV, θ is the UAV's heading, v is the UAV's velocity, h is the UAV's attitude, and u_1, u_2 are constrained by the dynamic capability of the UAVs namely the heading rate constraint and the acceleration constraint respectively.

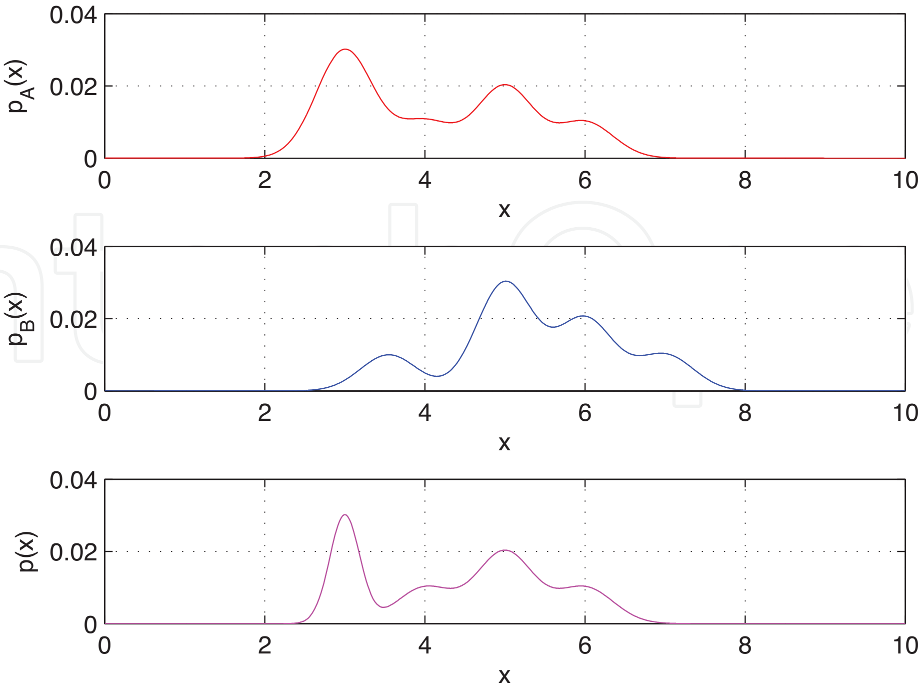


Fig. 8. Fusion of the probability density functions produced by the local particle filters

An inertial measurement unit (IMU) of a UAV usually consists of a three axis gyroscope and a three axis accelerometer. A vision sensor can be also mounted underneath the body of the UAV and is used to extract points of interest in the environment. The UAV also carries a barometric pressure sensor for aiding of the platform attitude estimation. A GPS sensor, can be also mounted on the board. The sensor data is filtered and fused to obtain estimates of the desired entities such as platform and feature position (Vissière et al. 2008).

5.2 Differential flatness for finite dimensional systems

Flatness-based control is proposed for steering the UAV along a desirable trajectory (Oriolo et al. 2002), (Villagra et al. 2007), (Fliess et al. 1999). The main principles of flatness- based control are as follows: A finite dimensional system is considered. This can be written in the general form of an ODE, i.e.

$$S_i(w, \dot{w}, \ddot{w}, \cdots, w^i), \;\; i = 1, 2, \cdots, q \tag{84}$$

The quantity w denotes the system variable while w^i, i = 1, 2, $\cdots$, q are its derivatives (these and can be for instance the elements of the system's state vector). The system of Eq. (1) is said to be differentially flat if there exists a collection of m functions $y = (y_1, \cdots, y_m)$ of the system variables w_i, $i = 1, \cdots, s$ and of their time-derivatives, i.e.

$$y_i = \phi(w, \dot{w}, \ddot{w}, \cdots, w^{\alpha_i}), i = 1, \cdots, m \tag{85}$$

such that the following two conditions are satisfied (Fliess et al. 1999), (Rigatos 2008):

1. There does not exist any differential relation of the form

$$R(y, \dot{y}, \cdots, y^{\beta}) = 0 \tag{86}$$

which implies that the derivatives of the flat output are not coupled in the sense of an ODE, or equivalently it can be said that the flat output is differentially independent.

2. All system variables, i.e. the components of w (elements of the system's state vectors) can be expressed using only the flat output y and its time derivatives

$$w_i = \psi_i(y,\dot{y},\cdots,y^{\gamma_i}), i = 1,\cdots,s \tag{87}$$

An equivalent definition of differentially flat systems is as follows:
Definition: The system $\dot{x} = f(x,u)$, $x \in R^n$, $u \in R^m$ is differentially flat if there exist relations h : $R^n \times R^m \rightarrow R^m$, ϕ: $(R^m)^r \rightarrow R^n$ and ψ: $(R^m)^{r+1} \rightarrow R^m$, such that $y = h(x,u,\dot{u},\cdots,u^{(r)})$, $x = \phi(y,\dot{y},\cdots,y^{(r-1)})$ and $u = \psi(y,\dot{y},...,y^{(r-1)},y^{(r)})$. This means that all system dynamics can be expressed as a function of the flat output and its derivatives, therefore the state vector and the control input can be written as $x(y) = \phi(y(t),\dot{y}(t),...,y^{(r)}(t))$ and $u = \psi(y(t),\dot{y}(t),...,y^{(r)}(t))$.
It is noted that for linear systems the property of differential flatness is equivalent to that of controllability.

5.3 Differential flatness of the UAV kinematic model

It is assumed that the helicopter-like UAV, performs manoeuvres at a constant altitude. Then, from Eq. (83) one can obtain the following description for the UAV kinematics

$$\dot{x} = vcos(\theta), \dot{y} = vsin(\theta), \dot{\theta} = \frac{v}{l}tan(\phi) \tag{88}$$

where using the analogous of the unicycle robot v is the velocity of the UAV, l is the UAV's length, θ is the UAV's orientation (angle between the transversal axis of the UAV and axis OX), and ϕ is a steering angle. The flat output is the cartesian position of the UAV's center of gravity, denoted as $\eta = (x,y)$, while the other model parameters can be written as:

$$v = \pm\|\dot{\eta}\| \quad \begin{pmatrix} cos(\theta) \\ sin(\theta) \end{pmatrix} = \frac{\dot{\eta}}{v} \quad tan(\phi) = ldet(\dot{\eta}\ddot{\eta}) / v^3 \tag{89}$$

These formulas show simply that θ is the tangent angle of the curve traced by P and $tan(\phi)$ is the associated curvature. With reference to a generic driftless nonlinear system

$$\dot{q}, q \in R^n, w \in R^m \tag{90}$$

dynamic feedback linearization consists in finding a feedback compensator of the form

$$\begin{aligned} \dot{\xi} &= \alpha(q,\xi) + b(q,\xi)u \\ w &= c(q,\xi) + d(q,\xi)u \end{aligned} \tag{91}$$

with state $\xi \in R^v$ and input $u \in R^m$, such that the closed-loop system of Eq. (90) and Eq. (91) is equivalent under a state transformation $z = T(q, \xi)$ to a linear system. The starting point is the definition of a m-dimensional output $\eta = h(q)$ to which a desired behavior can be assigned. One then proceeds by successively differentiating the output until the input appears in a non-singular way. If the sum of the output differentiation orders equals the dimension $n + v$ of the extended state space, full input-state-output linearization is obtained

(In this case η is also called a flat output). The closed-loop system is then equivalent to a set of decoupled input-output chains of integrators from u_i to η_i. The exact linearization procedure is illustrated for the unicycle model of Eq. (21). As flat output the coordinates of the center of gravity of the vehicle is considered $\eta = (x,y)$. Differentiation with respect to time then yields (Oriolo et al. 2002), (Rigatos 2008)

$$\dot{\eta} = \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} cos(\theta) & 0 \\ sin(\theta) & 0 \end{pmatrix} \cdot \begin{pmatrix} v \\ \omega \end{pmatrix} \tag{92}$$

showing that only v affects $\dot{\eta}$, while the angular velocity ω cannot be recovered from this first-order differential information. To proceed, one needs to add an integrator (whose state is denoted by ξ) on the linear velocity input

$$v = \xi, \quad \dot{\xi} = \alpha \Rightarrow \dot{\eta} = \xi \begin{pmatrix} cos(\theta) \\ sin(\theta) \end{pmatrix} \tag{93}$$

where α denotes the linear acceleration of the UAV. Differentiating further one obtains

$$\ddot{\eta} = \dot{\xi} \begin{pmatrix} cos(\theta) \\ sin(\theta) \end{pmatrix} + \xi\dot{\theta} \begin{pmatrix} sin(\theta) \\ cos(\theta) \end{pmatrix} = \begin{pmatrix} cos(\theta) & -\xi sin(\theta) \\ sin(\theta) & \xi cos(\theta) \end{pmatrix} \begin{pmatrix} \alpha \\ \omega \end{pmatrix} \tag{94}$$

and the matrix multiplying the modified input (α,ω) is nonsingular if $\xi \neq 0$. Under this assumption one defines

$$\begin{pmatrix} \alpha \\ \omega \end{pmatrix} = \begin{pmatrix} cos(\theta) & -\xi sin(\theta) \\ sin(\theta) & \xi cos(\theta) \end{pmatrix} \cdot \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \tag{95}$$

$\ddot{\eta}$ is denoted as

$$\ddot{\eta} = \begin{pmatrix} \ddot{\eta}_1 \\ \ddot{\eta}_2 \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = u \tag{96}$$

which means that the desirable linear acceleration and the desirable angular velocity can be expressed using the transformed control inputs u_1 and u_2. Then, the resulting dynamic compensator is (return to the initial control inputs v and ω)

$$\begin{gathered} \dot{\xi} = u_1 cos(\theta) + u_2 sin(\theta) \\ v = \xi \\ \omega = \frac{u_2 cos(\theta) - u_1 sin(\theta)}{\xi} \end{gathered} \tag{97}$$

Being $\xi \in R$, it is $n + v = 3 + 1 = 4$, equal to the output differentiation order in Eq. (29). In the new coordinates

$$\begin{gathered} z_1 = x \\ z_2 = y \\ z_3 = \dot{x} = \xi cos(\theta) \\ z_4 = \dot{y} = \xi sin(\theta) \end{gathered} \tag{98}$$

The extended system is thus fully linearized and described by the chains of integrators, in Eq. (29), and can be rewritten as

$$\begin{aligned} \ddot{z}_1 &= u_1 \\ \ddot{z}_2 &= u_2 \end{aligned} \tag{99}$$

The dynamic compensator of Eq. (97) has a potential singularity at $\xi = v = 0$, i.e. when the UAV is not moving, which is a case never met when the UAV is in flight. It is noted however, that the occurrence of such a singularity is structural for non-holonomic systems. In general, this difficulty must be obviously taken into account when designing control laws on the equivalent linear model.

A nonlinear controller for output trajectory tracking, based on dynamic feedback linearization, is easily derived. Assume that the UAV must follow a smooth trajectory $(x_d(t), y_d(t))$ which is persistent, i.e. for which the nominal velocity $v_d = (\dot{x}_d^2 + \dot{y}_d^2)^{\frac{1}{2}}$ along the trajectory never goes to zeros (and thus singularities are avoided). On the equivalent and decoupled system of Eq. (32), one can easily design an exponentially stabilizing feedback for the desired trajectory, which has the form

$$\begin{aligned} u_1 &= \ddot{x}_d + k_{p_1}(x_d - x) + k_{d_1}(\dot{x}_d - \dot{x}) \\ u_2 &= \ddot{y}_d + k_{p_1}(y_d - y) + k_{d_1}(\dot{y}_d - \dot{y}) \end{aligned} \tag{100}$$

and which results in the following error dynamics for the closed-loop system

$$\begin{aligned} \ddot{e}_x + k_{d_1}\dot{e}_x + k_{p_1}e_x &= 0 \\ \ddot{e}_y + k_{d_2}\dot{e}_y + k_{p_2}e_y &= 0 \end{aligned} \tag{101}$$

where $e_x = x - x_d$ and $e_y = y - y_d$. The proportional-derivative (PD) gains are chosen as $k_{p_1} > 0$ and $k_{d_1} > 0$ for $i = 1, 2$. Knowing the control inputs u_1, u_2, for the linearized system one can calculate the control inputs v and ω applied to the UAV, using Eq. (91). The above result is valid, provided that the dynamic feedback compensator does not meet the singularity. In the general case of design of flatness-based controllers, the following theorem assures the avoidance of singularities in the proposed control law (Oriolo et al. 2002):

Theorem: Let λ_{11}, λ_{12} and λ_{21}, λ_{22}, be respectively the eigenvalues of two equations of the error dynamics, given in Eq. (91). Assume that, for $i = 1{,}2$ it is $\lambda_{11} < \lambda_{12} < 0$ (negative real eigenvalues), and that λ_{i2} is sufficiently small. If

$$\min_{t \geq 0} \left\| \begin{pmatrix} \dot{x}_d(t) \\ \dot{y}_d(t) \end{pmatrix} \right\| \geq \begin{pmatrix} \dot{\epsilon}_x^0 \\ \dot{\epsilon}_y^0 \end{pmatrix} \tag{102}$$

with $\dot{\epsilon}_x^0 = \dot{\epsilon}_x(0) \neq 0$ and $\dot{\epsilon}_y^0 = \dot{\epsilon}_y(0) \neq 0$, then the singularity $\xi = 0$ is never met.

6. Simulation tests

6.1 Autonomous UAV navigation with Extended Information Filtering

It was assumed that $m = 2$ helicopter models were monitored by $n = 2$ different ground stations. At each ground station an Extended Kalman Filter was used to track each UAV. By fusing the measurements provided by the sensors mounted on each UAV, each local EKF was able to produce an estimation of a UAV's motion. Next, the state estimates obtained by the pair local EKFs associated with each UAV were fused with the use of the Extended Information Filter. This fusion-based state estimation scheme is depicted in Fig. 2. As explained in Section 2 the weighting of the state estimates of the local EKFs was performed using the local information matrices. The distributed fitering architecture is shown in Fig. 9.

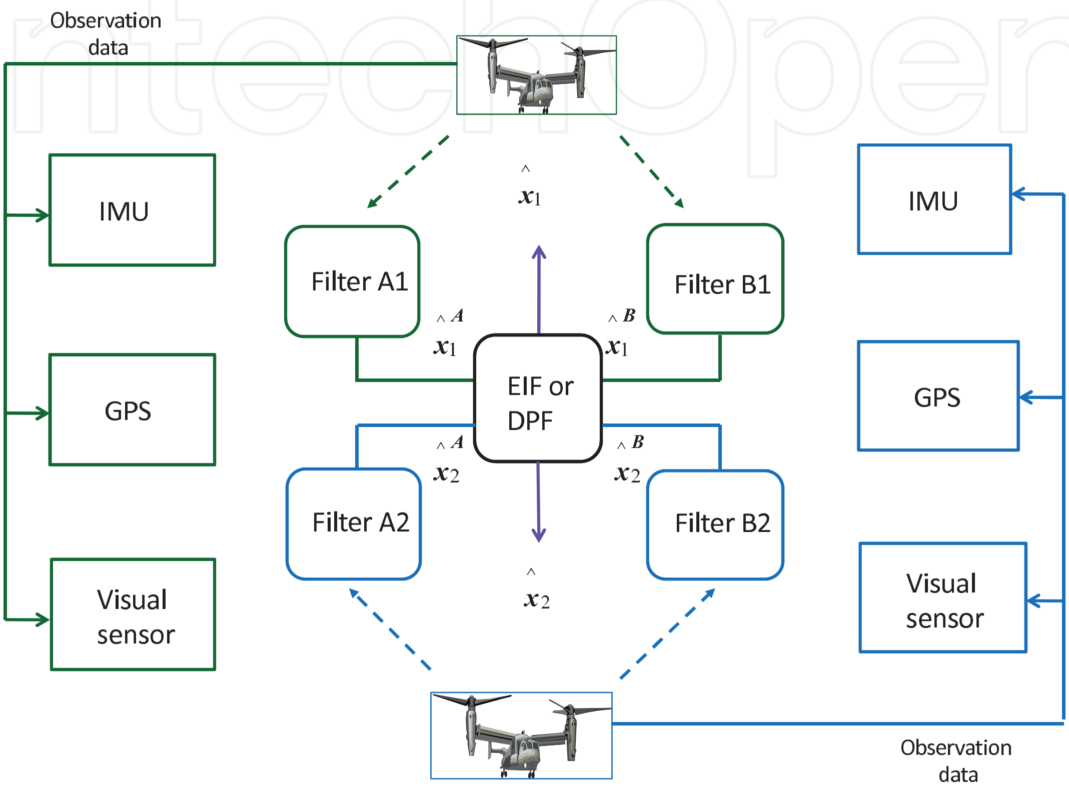


Fig. 9. Distributed Filtering over WSN

Next, some details will be given about the local EKF design for the UAV model of Eq. (88). The UAV's continuous-time kinematic equation is:

$$\dot{x}(t) = v(t)cos(\theta(t)),\ \dot{y}(t) = v(t)sin(\theta(t)),\ \dot{\theta}(t) = \omega(t) \tag{103}$$

The IMU system provides measurements or the UAV's position $[x,y]$ and the UAV's orientation angle θ over a sampling period T. These sensors are used to obtain an estimation of the displacement and the angular velocity of the UAV $v(t)$ and $\omega(t)$, respectively. The IMU sensors can introduce incremental errors, which result in an erroneous estimation of the orientation θ. To improve the accuracy of the UAV's localization, measurements from the GPS (or visual sensors) can be used. On the other hand, the GPS on this own is not always reliable since its signal can be intermittent. Therefore, to succeed accurate localization of the UAV it is necessary to fuse the GPS measurements with the IMU measurements of the UAV or with measurements from visual sensors (visual odometry).

The inertial coordinates system OXY is defined. Furthermore the coordinates system $O'X'Y'$ is considered (Fig. 10). $O'X'Y'$ results from OXY if it is rotated by an angle θ. The coordinates of the center of symmetry of the UAV with respect to OXY are (x,y), while the coordinates of

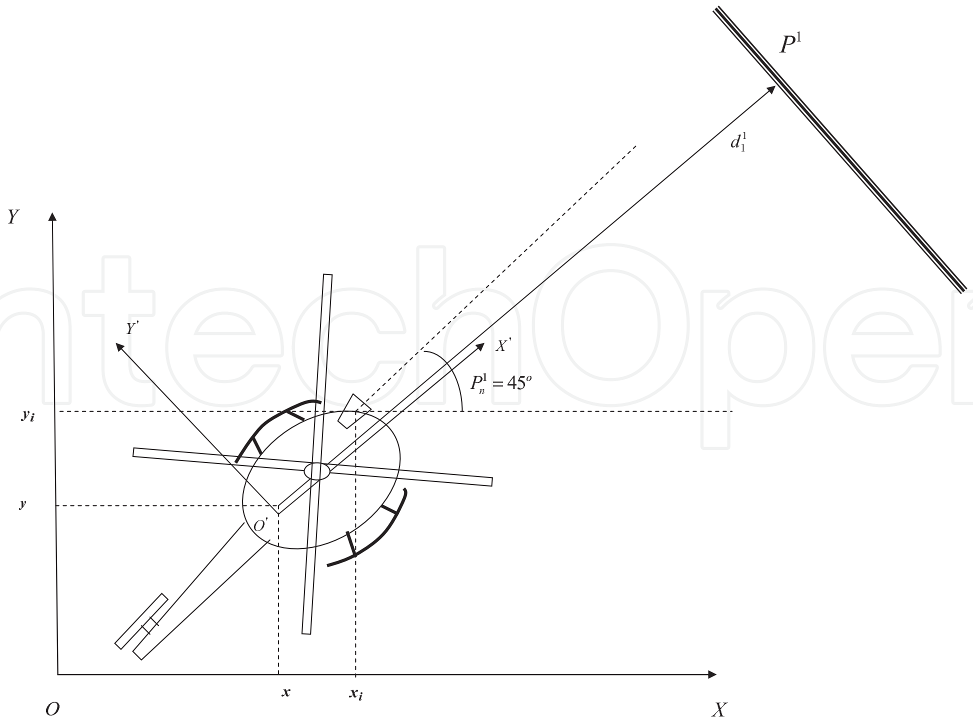


Fig. 10. Reference frames for the UAV

the GPS or visual sensor that is mounted on the UAV, with respect to $O'X'Y'$ are x_i', y_i'. The orientation of the GPS (or visual sensor) with respect to $OX'Y'$ is θ_i'. Thus the coordinates of the GPS or visual sensor with respect to OXY are (x_i,y_i) and its orientation is θ_i, and are given by:

$$\begin{aligned} x_i(k) &= x(k) + x_i' sin(\theta(k)) + y_i' cos(\theta(k)) \\ y_i(k) &= y(k) - x_i' cos(\theta(k)) + y_i' sin(\theta(k)) \\ \theta_i(k) &= \theta(k) + \theta_i \end{aligned} \tag{104}$$

For manoeuvres at constant altitude the GPS measurement (or the visual sensor measurement) can be considered as the measurement of the distance from a reference surface P^j. A reference surface P^j in the UAVs 2D flight area can be represented by P_r^j and P_n^j, where (i) P_r^j is the normal distance of the plane from the origin O, (ii) P_n^j is the angle between the normal line to the plane and the x-direction.

The GPS sensor (or visual sensor i) is at position $x_i(k)$,$y_i(k)$ with respect to the inertial coordinates system OXY and its orientation is $\theta_i(k)$. Using the above notation, the distance of the GPS (or visual sensor i), from the plane P^j is represented by P_r^j, P_n^j (see Fig. 10):

$$d_i^j(k) = P_r^j - x_i(k) cos(P_n^j) - y_i(k) sin(P_n^j) \tag{105}$$

Assuming a constant sampling period $\Delta t_k = T$ the measurement equation is $z(k+1) = \gamma(x(k)) + v(k)$, where $z(k)$ is the vector containing GPS (or visual sensor) and IMU measures and $v(k)$ is a white noise sequence $\sim N(0,R(kT))$.

By definition of the measurement vector one has that the output function is $\gamma(x(k)) = [x(k),y(k), \theta(k),d_1(k)]^T$. The UAV state is $[x(k),y(k), \theta(k)]^T$ and the control input is denoted by

$U(k) = [v(k),\omega(k)]^T$. To obtain the Extended Kalman Filter (EKF), the kinematic model of the UAV is linearized about the estimates $\hat{x}(k)$ and $\hat{x}^-(k)$ the control input $U(k-1)$ is applied. The *measurement update* of the EKF is

$$\begin{aligned}
&K(k) = P^-(k)J_\gamma^T(\hat{x}^-(k))[J_\gamma(\hat{x}^-(k))P^-(k)J_\gamma^T(\hat{x}^-(k)) + R(k)]^{-1}\\
&\hat{x}(k) = \hat{x}^-(k) + K(k)[z(k) - \gamma(\hat{x}^-(k))]\\
&P(k) = P^-(k) - K(k)J_\gamma^T P^-(k)
\end{aligned}$$

The *time update* of the EKF is

$$\begin{aligned}
&P^-(k+1) = J_\phi(\hat{x}(k))P(k)J_\phi^T(\hat{x}(k)) + Q(k)\\
&\hat{x}^-(k+1) = \phi(\hat{x}(k)) + L(k)U(k)
\end{aligned}$$

$$\text{where } L(k) = \begin{pmatrix} Tcos(\theta(k)) & 0\\ Tsin(\theta(k)) & 0\\ 0 & T \end{pmatrix} \text{and } J_\phi(\hat{x}(k)) = \begin{pmatrix} 1 & 0 & -v(k)sin(\theta)T\\ 0 & 1 & -v(k)cos(\theta)T\\ 0 & 0 & 1 \end{pmatrix} \tag{106}$$

while $Q(k) = diag[\sigma^2(k),\sigma^2(k),\sigma^2(k)]$, with $\sigma^2(k)$ chosen to be 10^{-3} and $\phi(\hat{x}(k)) = [\hat{x}(k),\hat{y}(k),\hat{\theta}(k)]^T$, $\gamma(\hat{x}(k)) = [\hat{x}(k),\hat{y}(k),\hat{\theta}(k),d(k)]^T$, i.e.

$$\gamma(\hat{x}(k)) = \begin{pmatrix} \hat{x}(k)\\ \hat{y}(k)\\ \hat{\theta}(k)\\ P_r^j - x_i(k))cos(P_n^j) - y_i(k)sin(P_n^j) \end{pmatrix} \tag{107}$$

In the calculation of the observation equation Jacobian one gets

$$J_\gamma^T(\hat{x}^-(k)) = \begin{pmatrix} 1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\\ -cos(P_n^j) & -sin(P_n^j) & \{x_i' cos(\theta - P_n^j) - y_i' sin(\theta - P_n^j)\} \end{pmatrix} \tag{108}$$

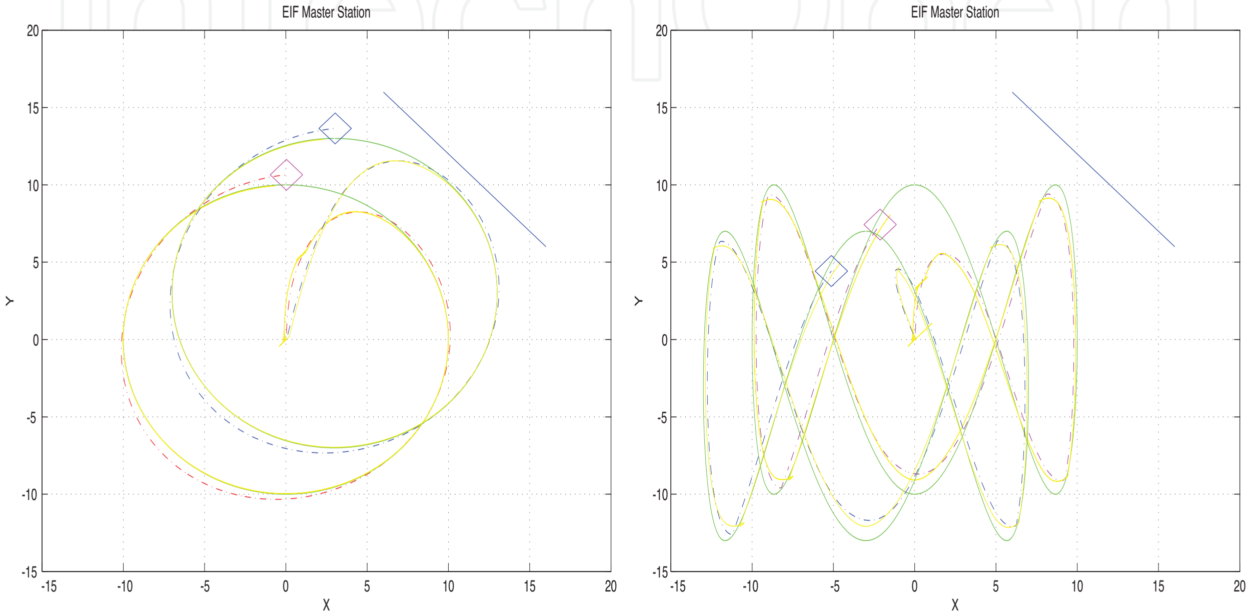
The UAV is steered by a dynamic feedback linearization control algorithm which is based the flatness-based control analyzed in Section 5:

$$\begin{aligned}
&u_1 = \ddot{x}_d + K_{p_1}(x_d - x) + K_{d_1}(\dot{x}_d - \dot{x})\\
&u_2 = \ddot{y}_d + K_{p_2}(y_d - y) + K_{d_2}(\dot{y}_d - \dot{y})\\
&\dot{\xi} = u_1 cos(\theta) + u_2 sin(\theta)\\
&v = \xi,\ \omega = \frac{u_2 cos(\theta) - u_1 sin(\theta)}{\xi}
\end{aligned} \tag{109}$$

Under the control law of Eq. (109) the dynamics of the tracking error finally becomes

$$\begin{aligned} \ddot{e}_x + K_{d_1}\dot{e}_x + K_{p_1}e_x = 0 \\ \ddot{e}_y + K_{d_2}\dot{e}_x + K_{p_2}e_y = 0 \end{aligned} \qquad (110)$$

where $e_x = x - xd$ and $e_y = y - y_d$. The proportional-derivative (PD) gains are chosen as K_{p_1} and K_{d_1}, for $i = 1, 2$.



(a) (b)

Fig. 11. Autonomous navigation of the multi-UAV system when the UAVs state vector is estimated with the use of the Extended Information Filter (a) tracking of circular reference trajectory (b) tracking of a curve-shaped reference trajectory

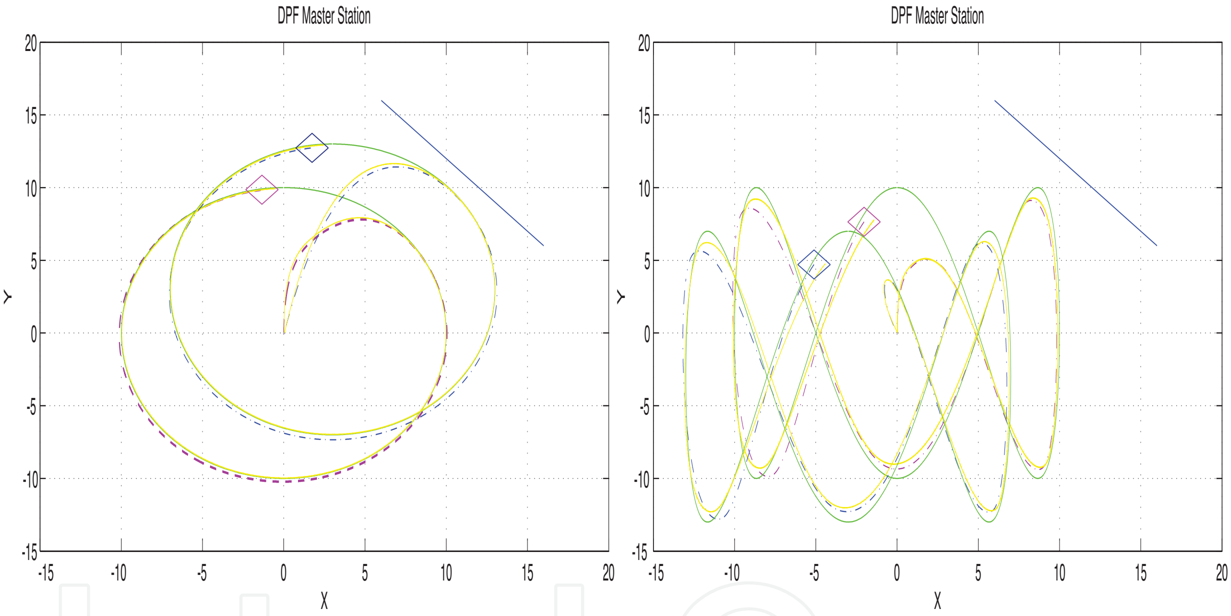
Results on the performance of the Extended Information Filter in estimating the state vectors of multiple UAVs when observed by distributed processing units is given in Fig. 11. Using distributed EKFs and fusion through the Extended Information Filter is more robust comparing to the centralized EKF since (i) if a local processing unit is subject to a fault then state estimation becomes is still possible and can be used for accurate localization of the UAV, as well as for tracking of desirable flight paths, (ii) communication overhead remains low even in the case of a large number of distributed measurement units, because the greatest part of state estimation is performed locally and only information matrices and state vectors are communicated between the local processing units, (iii) the aggregation performed on the local EKF also compensates for deviations in state estimates of local filters (which can be due to linearization errors).

6.2 Autonomous UAV navigation with Distributed Particle Filtering

Details on the implementation of the local particle filters are given first. Each local particle filter provides an estimation of the UAV's state vector using sensor fusion. The UAV model described in Eq. (103), and the control law given in Eq. (109) are used again.

The *measurement update* of the PF is $p(x(k)\,|\,Z)=\sum_{i=1}^{N} w_k^i \delta_{\xi_{k^-}^i}(x(k))$ with $w_k^i=\frac{w_{k^-}^i p(z(k)\,|\,x(k)=\xi_{k^-}^i)}{\sum_{j=1}^{N} w_k^j p(z(k)\,|\,x(k)=\xi_{k^-}^j)}$ where the measurement equation is given by $\hat{z}(k)=z(k)+v(k)$ with $z(k)$ = $[x(k), y(k), \theta(k), d(k)]^T$, and $v(k)$ =measurement noise.

The *time update* of the PF is $p(x(k+1)\,|\,Z)=\sum_{i=1}^{N} w_k^i \delta_{\xi_k^i}(x(k))$ where $\xi_k^i \sim p(x(k+1)\,|\,x(k)=\xi_{k^-}^i)$ and the state equation is $\hat{x}^-=\phi(x(k))+L(k)U(k)$, where $\phi(x(k))$, $L(k)$, and $U(k)$ are defined in subsection 6.1. At each run of the time update of the PF, the state vector estimation $\hat{x}^-(k+1)$ is calculated N times, starting each time from a different value of the state vector ξ_k^i. Although the Distributed Particle Filter can function under any noise distribution in the simulation experiments the measurement noise was assumed to be Gaussian. The obtained results are given in Fig. 12.



(a) (b)

Fig. 12. Autonomous navigation of the multi-UAV system when the UAVs state vector is estimated with the use of the Distributed Particle Filter (a) tracking of circular reference trajectory (b) tracking of a curve-shaped reference trajectory

In the simulation experiments it was observed that the Distributed Particle Filter, for N = 1000 particles, succeeded more accurate state estimation (smaller variance) than the EIF and consequently enables better tracking of the desirable trajectories by the UAVs. This improved performance of the DPF over the EIF is due to the fact that the local EKFs that constitute the EIF introduce cumulative errors due to the EKF linearization assumption (truncation of higher order terms in the Taylor expansion of Eq. (2) and Eq. (4)). Comparing to the Extended Information Filter, the Distributed Particle Filter demands more computation resources and its computation cycle is longer. However, the computation cycle of PF can be drastically reduced on a computing machine with a fast processor or with parallel processors (Míguez 2007). Other significant issues that should be taken into account in the design of the Distributed Particle Filter are the consistency of the fusion performed between the probability density functions of the local filters and the communication overhead between the local filters.

The simulation results presented in Fig. 12 show the efficiency of the Distributed Particle Filtering in providing accurate localization for the multi-UAV system, as well as for implementing state estimation-based control schemes. The advantages of using Distributed Particle Filtering are summarized as follows: (i) there is robust state estimation which is not constrained by the assumption of Gaussian noises. The fusion performed between the local probability density functions enables to remove outlier particles thus resulting in an aggregate state distribution that confines with accuracy the real state vector of each UAV. If a local processing unit (local filter) fails the reliability of the aggregate state estimation will be preserved (ii) computation load can be better managed comparing to a centralized particle filtering architecture. The greatest part of the necessary computations is performed at the local filters. Moreover the advantage of communicating state posteriors over raw observations is bandwidth efficiency, which is particularly useful for control over a wireless sensor network.

7. Conclusions

The paper has examined the problem of localization and autonomous navigation of a multi-UAV system based on distributed filtering over sensor networks. Particular emphasis was paid to distributed particle filtering since this decentralized state estimation approach is not constrained by the assumption of noise Gaussian distribution. It was considered that m UAV (helicopter) models are monitored by n different ground stations. The overall concept was that at each monitoring station a filter should be used to track each UAV by fusing measurements which are provided by various UAV sensors, while by fusing the state estimates from the distributed local filters an aggregate state estimate for each UAV should be obtained.

The paper proposed first the Extended Information Filter (EIF) and the Unscented Information Filter (UIF) as possible approaches for fusing the state estimates obtained by the local monitoring stations, under the assumption of Gaussian noises. It was shown that the EIF and UIF estimated state vector can be used by a flatness-based controller that makes the UAV follow the desirable trajectory. The Extended Information Filter is a generalization of the Information Filter in which the local filters do not exchange raw measurements but send to an aggregation filter their local information matrices (inverse covariance matrices which can be also associated to the Fisher Information matrices) and their associated local information state vectors (products of the local Information matrices with the local state vectors). In case of nonlinear system dynamics, such as the considered UAV models, the calculation of the information matrices and information state vectors requires the linearization of the local observation equations in the system's state space description and consequently the computation of Jacobian matrices is needed.

In the case of the Unscented Information Filter there is no linearization of the UAVs observation equation. However the application of the Information Filter algorithm is possible through an implicit linearization which is performed by approximating the Jacobian matrix of the system's output equation by the product of the inverse of the state vector's covariance matrix (Fisher information matrix) with the cross-covariance matrix between the system's state vector and the system's output. Again, the local information matrices and the local information state vectors are transferred to an aggregation filter which produces the global estimation of the system's state vector.

Next, the Distributed Particle Filter (DPF) was proposed for fusing the state estimates provided by the local monitoring stations (local filters). The motivation for using DPF was that it is well-suited to accommodate non-Gaussian measurements. A difficulty in implementing distributed particle filtering is that particles from one particle set (which correspond to a local particle filter) do not have the same support (do not cover the same area and points on the samples space) as particles from another particle set (which are associated with another particle filter). This can be resolved by transforming the particles set into Gaussian mixtures, and defining the global probability distribution on the common support set of the probability density functions associated with the local filters. Suitable importance resampling is proposed so as to derive the weights of the joint distribution after removing the common information contained in the probability density functions of the local filters. The state vector which is estimated with the use of the DPF was again used by the flatness-based controller to make each UAV follow a desirable flight path.

Comparing to centralized state estimation and control the proposed distributed state estimation and control schemes have significant advantages: (i) they are fault tolerant: if a local processing unit is subject to a fault then state estimation is still possible and accurate, (ii) the computation load is distributed between local processing units and since there is no need to exchange a large amount of information, the associated communication bandwidth is low. In the case of the Extended Information Filter and of the Unscented Information Filter the information transmitted between the local processing units takes the form of the information covariance matrices and the information state vectors. In the case of Distributed Particle Filtering the information transmitted between the local processing units takes the form of Gaussian mixtures. The performance of the Extended Information Filter and of the Distributed Particle Filter was evaluated through simulation experiments in the case of a 2-UAV model monitored and remotely navigated by two local stations.

Comparing the DPF to the EIF through simulation experiments it was observed that the Distributed Particle Filter, succeeded more accurate state estimation (smaller variance) than the EIF and consequently enabled better tracking of the desirable trajectories by the UAVs. This improved performance of the DPF over the EIF is explained according to to the fact that the local EKFs that constitute the EIF introduce cumulative errors due to the EKF linearization assumption. It was also observed that the Distributed Particle Filter demands more computation resources than the Extended Information Filter and that its computation cycle is longer. However, the computation cycle of the DPF can be drastically reduced on a computing machine with a fast processor or with parallel processors. Other issues that should be taken into account in the design of the Distributed Particle Filter are the consistency of the fusion performed between the probability density functions of the local filters and the communication overhead between the local filters.

8. References

[Beard et al. 2002] Beard, R.W.; McLain, T.W., Goodrich, M., & Anderson, E.P. (2002). Coordinated Target Assignment and Intercept for Unmanned Air Vehicles. *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 6, pp. 911-922, 2002.

[Caballero et al. 2008] Caballero, F.; Merino, L., Ferruz, J., & Ollero, A. (2008). A particle filtering method for Wireless Sensor Network localization with an aerial robot beacon. *Proc: IEEE International Conference on Robotics and Automation 2006*, pp. 2860-2865, 2008.

[Coué et al. 2003] Coué, C.; Pradalier, C. & Laugier, C. (2003). Bayesian Programming MultiTarget Tracking: an Automotive Application. *Int. Conf. on Field and Service Robotics*, Lake Yamanaka (Japan), July 2003.

[Coué et al. 2006] Coué, C; Pradalier, C., Laugier, C., Fraichard, T. & Bessiére, P. (2006). Bayesian Occupancy Filtering for Multitarget Tracking: An Automotive Application. *The International Journal of Robotics Research*, Vol. 25, No. 1, pp. 19-30, 2006.

[Deming & Perlovsky 2007] Deming, R.W. & Perlovsky, L.I. (2007). Concurrent multi-target localization, data association, and navigation for a swarm of flying sensors. *Information Fusion*, Elsevier, vol.8, no.3, pp. 316-330, 2007.

[Fliess et al. 1999] Fliess, M. & Mounier, H. (1999). *Tracking control and π-freeness of infinite dimensional linear systems, Dynamical Systems, Control, Coding and Computer Vision*, (G. Picci and D.S. Gilliam, Eds.), Vol. 258, pp. 41-68, Birkhaüser, 1999.

[Gan & Harris 2001] Gan Q. & Harris, C.J. (2001). Comparison of two measurement fusion methods for Kalman-filter-based multisensor data fusion. *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 37, No.1, pp. 273-280, 2001.

[Gao et al. 2009] Gao, S; Zhong, Y. Zhang, X., & Shirinzadeh, B. (2009). Multi-sensor optimal data fusion for INS/GPS/SAR integrated navigation system. *Aerospace Science and Technology*, Elsevier, Vol. 13, pp. 232-237, 2009.

[Hue et al. 2002] Hue, C.; Le Cadre, J.P. & P´erez, P. (2002). Tracking Multiple Objects with Particle Filtering. *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 38. No.3, pp. 791-812, 2002.

[Ing & Coates 2005] Ing, J. & Coates, M.G. (2005). Parallel particle filters for tracking in wireless sensor networks. *IEEE Workshop on Signal Processing Advances in Wireless Communications, SPAWC 2005*, art. no. 1506277, pp. 935-939, 2005.

[Julier et al. 2000] Julier, S.; Uhlmann, J. & Durrant-Whyte, H.F. (2000). A new method for the nonlinear transformations of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, Vol.45, No.3, pp. 477-482, 2000.

[Julier et al. 2004] Julier, S.J. & Uhlmann, J.K. (2004). Unscented Filtering and Nonlinear Estimation, *Proceedings of the IEEE*, Vol.92, pp. 401-422, 2004.

[Léchevin & Rabbath 2006] Léchevin, N. & Rabbath, C.A. (2006). Sampled-data control of a class of nonlinear flat systems with application to unicycle trajectory tracking. *ASME Journal of Dynamical Systems, Measurement and Control*, vol. 128, No.3, pp. 722-728, 2006.

[Lee et al. 2008] Lee D.J. (2008). Unscented Information Filtering for Distributed Estimation and multiple sensor fusion. *AIAA Guidance, Navigation and Control Conference and Exhibit*, Aug. 2008, Hawai, USA.

[Lee et al. 2008] Lee, D.J. (2008). Nonlinear estimation and multiple sensor fusion using unscented information filtering. *IEEE Signal Processing Letters*, Vol. 15, pp. 861-864, 2008.

[Mahler 2007] Mahler, R.P.S. (2003). *Statistical Multisource-Multitarget Information Fusion*. Artech House Inc. 2007.

[Makarenko & Durrant-Whyte 2006] Makarenko, A. & Durrant-Whyte, H. (2006). Decentralized Bayesian algorithms for active sensor networks. *Information Fusion*, Elsevier, Vol.7, pp. 418-433, 2006.

[Manyika & H. Durrant-Whyte 1994] Manyika, J. & Durrant-Whyte, H. (1994). *Data fusion and sensor management: a decentralized information theoretic approach*. Englewood Cliffs, NJ, Prentice Hall, 1994.

[Míguez 2007] Míguez, J. (2007). Analysis of parallelizable resampling algorithms for particle filtering. *Signal Processing*, Elsevier, Vol. 87, 3155-3174, 2007.

[Morelande & D. Mušicki 2005] Morelande, M.R. & Mušicki, D. (2005). Fast multiple target tracking using particle filters. *Proc. of the 44th IEEE Conference on Decision and Control, and the European Control Conference* 2005, Seville, Spain, December 12-15, 2005.

[Musso et al. 2001] Musso, C.; Oudjane, N. & Le Gland, F. (2001). *Imrpoving regularized particle filters, In: Sequential Monte Carlo Methods in Practice*. A. Doucet, N. de Freitas and N. Gordon Eds., Springer-Verlag 2001, pp. 247-272, 2001.

[Nettleton et al. 2003] Nettleton, E.; Durrant-Whyte & H. Sukkarieh, S. (2003). A robust architecture for decentralized data fusion. *ICAR03, 11th International Conference on Advanced Robotics*, Coimbra, Portugal, 2003.

[Olfati-Saber 2005] Olfati-Saber, R. (2005). Distributed Kalman Filter with Embedded Consensus Filters. *Proc. 44th IEEE Conference on Decision and Control*, pp. 8179-8184, Seville, Spain, 2005.

[Olfati-Saber 2006] Olfati-Saber, R. (2006). Distributed Kalman Filtering and Sensor Fusion in Sensor Networks. *Lecture notes in control and information sciences*, Vol. 331, pp. 157-167.

[Ong et al. 2006] Ong, L.L.; Upcroft, B., Bailey, T., Ridley, M. Sukkarieh, S. & Durrant-Whyte, H. (2006). A decentralized particle filtering algorithm for multi-target tracking across multiple flight vehicles. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, October 2006.

[Ong et al. 2008] Ong, L.L.; Bailey, T., Durrant-Whyte, H. & Upcroft, B. (2008). Decentralized Particle Filtering for Multiple Target Tracking in Wireless Sensor Networks. *Fusion 2008, The 11th International Conference on Information Fusion*, Cologne, Germany, July 2008.

[Oriolo et al. 2002] Oriolo, G.; De Luca, A. & Vendittelli, M. (2002). WMR Control Via Dynamic Feedback Linearization: Design, Implementation and Experimental Validation. *IEEE Transactions on Control Systems Technology*, Vol. 10, No.6, pp. 835-852, 2002.

[Ren & Beard 2004] Ren, W. & Beard, R.W. (2004). Trajectory tracking for unmanned air vehicles with velocity and heading rate constraints. *IEEE Transactions on Control Systems Technology*, Vol. 12, No. 5, pp. 706-716, 2004.

[Rigatos & Tzafestas 2007] Rigatos, G.G. & Tzafestas, S.G. (2007). Extended Kalman Filtering for Fuzzy Modeling and Multi-Sensor Fusion. *Mathematical and Computer Modeling of Dynamical Systems*, Vol. 13, No 3, Taylor and Francis, 2007.

[Rigatos 2008] Rigatos, G.G. (2008). Autonomous robots navigation using flatness-based control and multi-sensor fusion. *Robotics, Automation and Control*, (P. Pecherkova, M. Fliidr and J. Dunik, Eds), I-Tech Education and Publishing KG, Vienna Austria, pp. 394-416, 2008.

[Rigatos 2009a] Rigatos, G.G. (2009). Particle Filtering for State Estimation in Nonlinear Industrial Systems. *IEEE Transactions on Instrumentation and Measurement*, Vol. 58, No. 11, pp. 3885-3900, 2009.

[Rigatos 2009b] Rigatos, G.G. (2009). Sigma-point Kalman Filters and Particle Filters for integrated navigation of unmanned aerial vehicles. *Intl. Workshop on Robotics for Risky Interventions and Environmental Surveillance, RISE 2009*, Brussels, Belgium, Jan. 2009.

[Rigatos & Zhang 2009] Rigatos, G. & Zhang, Q. (2009). Fuzzy model validation using the local statistical approach. *Fuzzy Sets and Systems*, Elsevier, Vol. 60, No.7, pp. 882-904, 2009.

[Rosencrantz et al. 2003] Rosencrantz, M.; Gordon, G., Thrun, S. (2003). Decentralized data fusion with distributed particle filtering. *Proceedings of the Conference of Uncertainty in AI (UAI)*, Acapulco, Mexico, 2003.

[Särrkä 2007] Särrkä, S. (2007). On Unscented Kalman Filtering for state estimation of continuous-time nonlinear systems. *IEEE Transactions on Automatic Control*, Vol.52, No.9, pp.1631-1641, 2007.

[Shima et al. 2007] Shima, T.; Rasmussen, S.J. & Chandler, P. (2007). UAV team decision and control using efficient collaborative estimation. *Journal of Dynamic Systems, Measurement and Control*, Transactions of the ASME, Vol. 129, No 5, pp. 609-619, 2007.

[Singh & Fuller 2001] Singh, L. & Fuller, J. (2001). Trajectory generation for a UAV in urban terrain using nonlinear MPC. *Proceedings of American Control Conference*, pp. 2301-2308, 2001.

[Thrun et al. 2005] Thrun, S.; Burgard, M. & Fox, D. (2005). *Probabilistic Robotics*, MIT Press, 2005.

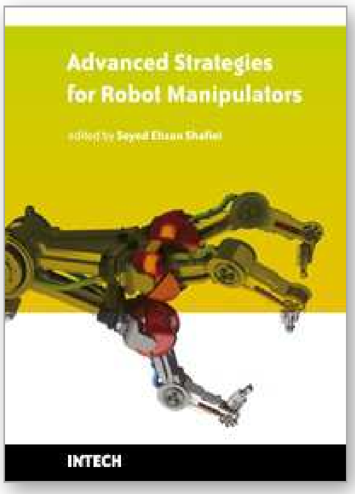
[Vercauteren &Wang 2005] Vercauteren, T. &Wang, X. (2005). Decentralized Sigma-Point Information Filters for Target Tracking in Collaborative Sensor Networks. *IEEE Transactions on Signal Processing*, Vol.53, No.8, pp. 2997-3009, 2005.

[Villagra et al. 2007] Villagra, J.; d'Andrea-Novel, B., Mounier, H. & Pengov, M. (2007). Flatness-based vehicle steering control strategy with SDRE feedback gains tuned via a sensitivity approach. emphIEEE Transactions on Control Systems Technology, Vol. 15, pp. 554-565, 2007.

[Vissière et al. 2008] Vissière, D.; Bristeau, P.-J., Martin, A.P. & N. Petit (2008). Experimental autonomous flight of a small-scaled helicopter using accurate dynamics model and low-cost sensors. *Proceedings of the 17thWorld Congress The International Federation of Automatic Control Seoul*, Korea, July 2008.

[Watanabe & Tzafestas 1992] Watanabe, K. & Tzafestas, S.G. (1992). Filtering, Smoothing and Control in Discrete-Time Stochastic Distributed-Sensor Networks, *In: Stochastic Large- Scale Engineering Systems* (S.G. Tzafestas and K. Watanabe Eds), pp. 229-252, Marcel Dekker, 1992.

[Zhang et al. 2005] Zhang, Campillo, Q.F., C´erou, F. & F. Legland (2005). Nonlinear fault detection and isolation based on bootstrap particle filters, *Proc. of the 44th IEEE Conference on Decision and Control, and European Control Conference*, Seville Spain, Dec. 2005.


Advanced Strategies for Robot Manipulators
edited by Seyed Ehsan Shafiei
INTECH

Advanced Strategies for Robot Manipulators
Edited by S. Ehsan Shafiei

ISBN 978-953-307-099-5
Hard cover, 428 pages
Publisher Sciyo
Published online 12, August, 2010
Published in print edition August, 2010

Amongst the robotic systems, robot manipulators have proven themselves to be of increasing importance and are widely adopted to substitute for human in repetitive and/or hazardous tasks. Modern manipulators are designed complicatedly and need to do more precise, crucial and critical tasks. So, the simple traditional control methods cannot be efficient, and advanced control strategies with considering special constraints are needed to establish. In spite of the fact that groundbreaking researches have been carried out in this realm until now, there are still many novel aspects which have to be explored.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821