Performance Comparison of the Distributed Extended Kalman Filter and Markov Chain Distributed Particle Filter (MCDPF) ⋆

Sun Hwan Lee * **Matthew West** **

* *Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305 USA (e-mail: sunhlee@stanford.edu).*
** *Department of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA (e-mail: mwest@illinois.edu)*

Abstract: We compare the performance of two distributed nonlinear estimators for a multi-vehicle flocking system using range measurements only. The estimators are the Distributed Extended Kalman Filter (DEKF) and the Markov Chain Distributed Particle Filter (MCDPF), where the distributed implementation in both cases is done using consensus-type algorithms. The performance of the estimators is compared as the system complexity (number of vehicles) and measurement frequency are varied. It is shown that for simple systems (few vehicles) or high measurement frequency the DEKF method has lower expected error than MCDPF, while for complex systems (many vehicles) or low measurement frequency the MCDPF method is both more robust and more accurate.

Keywords: Target tracking filters, Estimation algorithm, Distributed estimation, Sensor network, Flocking model.

1. INTRODUCTION

Distributed filters are very useful tools to estimate the state of a system from measurements made by a sensor network. They arise in many areas, including fault detection (J. Chen and Somani, 2006), target tracking (Sheng et al., 2005) and surveillance (Yan et al., 2003), to mention only a few. A Distributed Kalman Filter (DKF) can be formulated using consensus techniques (Olfati-Saber, 2007; Olfati-Saber and Shamma, 2005) and the Extended Kalman Filter (EKF) is easily distributed using the same idea. Meanwhile, past work on distributed particle filters can be broadly categorized into two approaches, namely message passing approaches and consensus-based local information exchange methods. Message passing approaches transfer information along a predetermined route covering an entire network. For example, Coates (2004) passes parameters of the parametric model of the posterior distribution while Sheng et al. (2005) transmits the raw information (particles and weights) or the parameters of a Gaussian Mixture Model (GMM) approximation of the posterior distribution. Consensus based methods communicate the information only between the nearest nodes and achieve global consistency by consensus filtering. The type of exchanged information can be, for example, the parameters of a GMM approximation (Gu, 2007) or the local mean and covariance (Gu et al., 2008).

In Lee and West (2009) the Markov Chain Distributed Particle Filter (MCDPF) is proposed and is shown to be both robust and scalable. Furthermore, the MCDPF is able to deal with general nonlinear state estimation problems where each node measures insufficient information to estimate the state alone, such as bearing-only or range-only information. The MCDPF method transmits particles and associated weights according to a Markov chain random walk on a sensor network graph, updating the weights in exponential proportion to the number of visits to each node. It has been shown (Lee and West, 2009) that the MCDPF converges to the Centralized Particle Filter (CPF) in the limit of fast communication, but a comparison of performance with existing methods has not been performed. In general, particle filters are superior to an EKF when the observed system is highly nonlinear, even to the extent that the EKF may be divergent while a particle filter gives usable results. In this paper, the performance of the MCDPF and a Distributed EKF (DEKF) are compared for a highly nonlinear system, namely a flocking model, and the regimes in which each type of filter is superior are investigated.

The contents of this paper are as follows. In section 2, a review of the EKF, DEKF, and consensus schemes is provided. Section 3 contains the background of the CPF and MCDPF algorithms. Numerical results for the example system, range-only tracking problem for a flocking model with 4 and 10 vehicles, is given in section 4 and conclusions are made in section 5.

2. EXTENDED KALMAN FILTER

In this section the Extended Kalman Filter (EKF) (Anderson and Moore, 1979) in information filter form is reviewed and the distributed EKF using a consensus scheme is in-

⋆ Sun Hwan Lee is supported by the Samsung Scholarship foundation.

troduced. Suppose we have the following nonlinear system with Gaussian noise.

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t) + q_t \tag{1}$$
$$\mathbf{z}_t = h(\mathbf{x}_t) + r_t. \tag{2}$$

where $\mathbf{x}_t \in \mathbf{R}^n$, $\mathbf{z}_t \in \mathbf{R}^p$, and $w_t \sim \mathcal{N}(0, Q_t), r_t \sim \mathcal{N}(0, R_t)$ are process and measurement noises respectively. The following notations are used for state estimation.

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{E}(\mathbf{x}_t|\mathbf{z}_{1:t-1}), \quad \hat{\mathbf{x}}_{t|t} = \mathbf{E}(\mathbf{x}_t|\mathbf{z}_{1:t}). \tag{3}$$

Here $\mathbf{z}_{1:t}$ is the time series of measurements, $\mathbf{z}_1, \ldots, \mathbf{z}_t$.

2.1 Centralized Approach

The EKF estimates the state of a nonlinear system by linearizing the system and measurement equations around the previously estimated state. Given the probability distribution of the initial state, $\mathbf{x}_0 \sim \mathcal{N}(\bar{\mathbf{x}}_0, P_0)$, the EKF initializes the information matrix and information state as follows.

$$Y_{0|0} = P_0^{-1}, \quad \mathbf{y}_{0|0} = Y_{0|0}\bar{x}_0, \quad \hat{\mathbf{x}}_{0|0} = Y_{0|0}^{-1}\mathbf{y}_{0|0}. \tag{4}$$

For the prediction step,

$$Y_{t|t-1} = \left(F_{t-1}Y_{t-1|t-1}^{-1}F_{t-1}^T + Q_t\right)^{-1} \tag{5}$$
$$\mathbf{y}_{t|t-1} = Y_{t|t-1}f(\hat{\mathbf{x}}_{t-1|t-1}) \tag{6}$$
$$\hat{\mathbf{x}}_{t|t-1} = Y_{t|t-1}^{-1}\mathbf{y}_{t|t-1}. \tag{7}$$

The matrix F_{t-1} is Jacobian of the function $f(\mathbf{x}_t)$ evaluated at $\hat{\mathbf{x}}_{t-1|t-1}$.

$$F_{t-1} = \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{t-1|t-1}}. \tag{8}$$

For the measurement step,

$$Y_{t|t} = Y_{t|t-1} + H_t^T R_t^{-1} H_t \tag{9}$$
$$\mathbf{y}_{t|t} = \mathbf{y}_{t|t-1} + H_t^T R_t^{-1}\tilde{\mathbf{z}}_t \tag{10}$$
$$\hat{\mathbf{x}}_{t|t} = Y_{t|t}^{-1}\mathbf{y}_{t|t} \tag{11}$$

where $\tilde{\mathbf{z}}_t = \mathbf{z}_t - h(\hat{\mathbf{x}}_{t|t-1}) + H_t\hat{\mathbf{x}}_{t|t-1}$ and, similarly, H_t is the Jacobian of the measurement equation $h(\mathbf{x}_t)$,

$$H_t = \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{t|t-1}}. \tag{12}$$

2.2 Distributed Approach

In the distributed approach we assume there are m nodes measuring the partial observations independently as follows.

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t) + q_t \tag{13}$$
$$\begin{bmatrix}\mathbf{z}_{1,t}\\ \vdots \\ \mathbf{z}_{m,t}\end{bmatrix} = \begin{bmatrix}h_1(\mathbf{x}_t) + r_{1,t}\\ \vdots \\ h_m(\mathbf{x}_t) + r_{m,t}\end{bmatrix}. \tag{14}$$

Here $\mathbf{z}_{i,t} \in \mathbf{R}^{p_i}$ with $\sum_{i=1}^m p_i = p$ and subscript i is the node index. In addition, the measurement noise at each node is assumed to be uncorrelated with that at other nodes, $\mathbf{E}[r_t r_t^T] = \mathrm{diag}(R_{1,t}, R_{2,t}, \ldots, R_{m,t})$. The initialization and prediction steps are the same as for the centralized approach, except for the fact that the same procedure is repeated for every node. The measurement step is distributed for $i = 1, \ldots, m$,

$$Y_{i,t|t} = Y_{i,t|t-1} + H_t^T R_t^{-1} H_t \tag{15}$$
$$= Y_{i,t|t-1} + \sum_{i=1}^m H_{i,t}^T R_{i,t}^{-1} H_{i,t} \tag{16}$$
$$\mathbf{y}_{i,t|t} = \mathbf{y}_{i,t|t-1} + H_t^T R_t^{-1}\tilde{\mathbf{z}}_t \tag{17}$$
$$= \mathbf{y}_{i,t|t-1} + \sum_{i=1}^m H_{i,t}^T R_{i,t}^{-1}\tilde{\mathbf{z}}_{i,t} \tag{18}$$
$$\hat{\mathbf{x}}_{i,t|t} = Y_{i,t|t}^{-1}\mathbf{y}_{i,t|t}. \tag{19}$$

Again $\tilde{\mathbf{z}}_{i,t} = \mathbf{z}_{i,t} - h_i(\hat{\mathbf{x}}_{i,t|t-1}) + H_{i,t}\hat{\mathbf{x}}_{i,t|t-1}$.

2.3 Consensus

Consensus is an iterative process for multiple agents to reach a common value, the average for example. A sensor network system can be modeled as a graph, $G = (V, E)$, with a normalized adjacency matrix $\mathcal{A}$. The vertices $V = \{1, \ldots, m\}$ correspond to nodes or sensors in the network, while edges E represent the connections between sensors. The neighbors of node i are given by $N_i = \{j \in V : \mathcal{A}_{ij} \neq 0\}$. The Distributed Kalman Filter (DKF) (Olfati-Saber, 2007; Olfati-Saber and Shamma, 2005) is implemented with a consensus scheme and the DEKF is able to be distributed in exactly the same way as the DKF if system and measurement matrices are replaced by the Jacobian matrices $F_{i,t}$ and $H_{i,t}$. When new measurements arrive at time t, the following terms are initialized.

$$S_{i,t} = H_{i,t}^T R_{i,t}^{-1} H_{i,t}, \quad s_{i,t} = H_{i,t}^T R_{i,t}^{-1}\tilde{\mathbf{z}}_{i,t}. \tag{20}$$

Then the average values of $S_{i,t}$ and $s_{i,t}$ are obtained by the following iterative consensus scheme (Xiao et al., 2006).

$$S_{i,t} \leftarrow S_{i,t} + \sum_{j\in N_i(t)} w_{ij}(t)\left(S_{j,t} - S_{i,t}\right) \tag{21}$$
$$s_{i,t} \leftarrow s_{i,t} + \sum_{j\in N_i(t)} w_{ij}(t)\left(s_{j,t} - s_{i,t}\right). \tag{22}$$

While many weight choices are possible, we consider the Metropolis weights,

$$w_{ij}(t) = \frac{1}{1 + \max\{d_i(t), d_j(t)\}} \tag{23}$$

where $d_i(t)$ is the degree of node i. The average values of $S_{i,t}$ and $s_{i,t}$ then converge exponentially to the true values. Then we can write the measurement step after the consensus procedure as

$$Y_{i,t|t} = Y_{i,t|t-1} + H_t^T R_t^{-1} H_t = Y_{i,t|t-1} + mS_{i,t} \tag{24}$$
$$\mathbf{y}_{i,t|t} = \mathbf{y}_{i,t|t-1} + H_t^T R_t^{-1}\tilde{\mathbf{z}}_t = \mathbf{y}_{i,t|t-1} + ms_{i,t}. \tag{25}$$

3. PARTICLE FILTERS

Given the nonlinear system (1–2) the filtering problem is to estimate the probability distribution of the true state $\mathbf{x}_t$ at time t given the time series of observations $\mathbf{z}_{1:t}$. The prediction and updating of the optimal filter based on Bayes' recursion are given as follows (Crisan and Doucet, 2002).

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) = \int_{\mathbf{R}^n} p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})K(\mathbf{x}_t|\mathbf{x}_{t-1})\,d\mathbf{x}_{t-1} \tag{26}$$
$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) = \frac{\rho(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{z}_{1:t-1})}{\int_{\mathbf{R}^n}\rho(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{z}_{1:t-1})\,d\mathbf{x}_t}. \tag{27}$$

where $K(\mathbf{x}_t|\mathbf{x}_{t-1})d\mathbf{x}_{t-1}$ is the probability transition kernel and $\rho(\mathbf{z}_t|\mathbf{x}_t)$ is the transition probability density of a measurement $\mathbf{z}_t$ given the state $\mathbf{x}_t$. Analytic solutions for the posterior distribution in (27) do not generally exist except in special cases, such as linear systems with Gaussian noise.

3.1 Centralized Approach

Particle filtering is a recursive method to estimate the true state, given the time series of measurements (Crisan and Doucet, 2002; Doucet et al., 2001). Suppose the posterior distribution at time $t-1$, $\pi_{t-1|t-1}(d\mathbf{x}_{t-1})$, is approximated by N particles $\{\mathbf{x}_{t-1}^i\}_{i=1}^N$. Then we have

$$p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}) \triangleq \pi_{t-1|t-1}(d\mathbf{x}_{t-1}) \tag{28}$$
$$\approx \pi_{t-1|t-1}^N(d\mathbf{x}_{t-1}) = \frac{1}{N}\sum_{i=1}^N \delta_{\mathbf{x}_{t-1}^i}(d\mathbf{x}_{t-1}).$$

where particle i is at position $\mathbf{x}_{t-1}^i$ in state space and $\delta_{\mathbf{x}_{t-1}^i}(d\mathbf{x}_{t-1})$ denotes a Dirac delta function located at $\mathbf{x}_{t-1}^i$. Now, particles go through the prediction and measurement update steps to approximate the posterior distribution at time t. Given N particles, new particles are sampled from the transition kernel density, $\tilde{\mathbf{x}}_t^i \sim \frac{1}{N}\sum_{i=1}^N K(\mathbf{x}_t|\mathbf{x}_{t-1}^i)$. This set of particles is the approximation of $\pi_{t|t-1}$,

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) \triangleq \pi_{t|t-1}(d\mathbf{x}_t) \tag{29}$$
$$\approx \tilde{\pi}_{t|t-1}^N(d\mathbf{x}_t) = \frac{1}{N}\sum_{i=1}^N \delta_{\tilde{\mathbf{x}}_t^i}(d\tilde{\mathbf{x}}_t). \tag{30}$$

If the empirical distribution in (30) is substituted in (27), we have the following distribution approximating the posterior distribution $p(\mathbf{x}_t|\mathbf{z}_{1:t})$.

$$\tilde{\pi}_{t|t}^N(d\mathbf{x}_t) \triangleq \frac{\rho(\mathbf{z}_t|\mathbf{x}_t)\tilde{\pi}_{t|t-1}^N(d\mathbf{x}_t)}{\int_{\mathbb{R}^n}\rho(\mathbf{z}_t|\mathbf{x}_t)\tilde{\pi}_{t|t-1}^N(d\mathbf{x}_t)\,d\mathbf{x}_t} \tag{31}$$
$$= \frac{\sum_{i=1}^N \rho(\mathbf{z}_t|\tilde{\mathbf{x}}_t^i)\delta_{\tilde{\mathbf{x}}_t^i}(d\tilde{\mathbf{x}}_t)}{\sum_{i=1}^N \rho(\mathbf{z}_t|\tilde{\mathbf{x}}_t^i)} = \sum_{i=1}^N w_t^i \delta_{\tilde{\mathbf{x}}_t^i}(d\tilde{\mathbf{x}}_t),$$

where $\sum_{i=1}^N w_t^i = 1$ and w_t^i are called the importance weights. To avoid degeneracy problems, particles are selected according to a resampling step that samples N particles from the empirical distribution, $\tilde{\pi}_{t|t}^N(d\mathbf{x}_t)$, and resets the weights to $\frac{1}{N}$. We then have an empirical distribution approximating posterior at time t given by

$$\pi_{t|t}^N(d\mathbf{x}_t) = \frac{1}{N}\sum_{i=1}^N \delta_{\mathbf{x}_t^i}(d\mathbf{x}_t). \tag{32}$$

3.2 Distributed Approach

The Markov Chain Distributed Particle Filter (MCDPF) and the theoretical analysis of convergence to the CPF are introduced in Lee and West (2009). The MCDPF distributes the CPF by exchanging raw data, consisting of particles and weights, according to a Markov chain random walk on the sensor network graph. Given a sensor network model (13-14), an uncorrelated noise structure enables us to have conditionally independent measurements at each node, $z_{i,t}$, given the true state $\mathbf{x}_t$. As a consequence of this assumption, the function $\rho(\mathbf{z}_t|\mathbf{x}_t^i)$ in (27) can be factorized by a product of $\rho_j(\mathbf{z}_{j,t}|\mathbf{x}_t^i)$ at each node,

$$\rho(\mathbf{z}_t|\mathbf{x}_t^i) = \prod_{j=1}^m \rho_j(\mathbf{z}_{j,t}|\mathbf{x}_t^i). \tag{33}$$

The MCDPF moves particles around the network according to the Markov chain on the network defined by using the normalized adjacency matrix $\mathcal{A}$ to compute the importance weights. The main idea is that each particle gains $\rho_i(\mathbf{z}_{i,t}|\mathbf{x}_t)$ exponentially proportional to the expected number of visits to node i. Suppose we have the graph $G = (V, E)$ based on the sensor network and the normalized adjacency matrix $\mathcal{A}$. In the MCDPF setting, the Markov chain is run for k steps on every particle after the prediction step and the number of visits to the i-th node is defined by $M(i,k)$. Considering the number of visits to each node, each particle multiplies $\rho_i(\mathbf{z}_{i,t}|\mathbf{x}_t)^{\frac{2|E(G)|}{kd(i)}}$ by its previous weight every time it visits the i-th node. If we have N particles after k Markov chain steps at a node, then the posterior distribution of the MCDPF is given as follows.

$$\tilde{\pi}_{t|t,k}^N(d\mathbf{x}_t) = \frac{\sum_{i=1}^N \prod_{j=1}^m \rho_j(\mathbf{z}_{j,t}|\tilde{\mathbf{x}}_t^i)^{\frac{2|E(G)|}{kd(j)}\times M(j,k)}\delta_{\tilde{\mathbf{x}}_t^i}(d\tilde{\mathbf{x}}_t)}{\sum_{i=1}^N \prod_{j=1}^m \rho_j(y_{j,t}|\tilde{\mathbf{x}}_t^i)^{\frac{2|E(G)|}{kd(j)}\times M(j,k)}}$$
$$= \sum_{i=1}^N w_{t,k}^i \delta_{\tilde{\mathbf{x}}_t^i}(d\tilde{\mathbf{x}}_t). \tag{34}$$

The MCDPF is defined in algorithm 1 below. We use the notation $\mathbf{x}_{j,t}^i$ for the i-th particle at node j at time t and $N(j)$ for the number of particle at node j. Also $\mathcal{I}_{i\to j}$ is the set of indices of particles moving from node i to j in the current Markov chain step and we recall that $\mathcal{A}$ is the adjacency matrix of the network.

Algorithm 1 Markov Chain Distributed Particle Filter (MCDPF)

Initialization:
$\{\mathbf{x}_{j,0}^i\}_{i=1}^N \sim p(\mathbf{x}_0)$, $\{w_{j,0}^i\}_{i=1}^N = \frac{1}{N}$ for $j = 1, \ldots, m$
Importance Sampling: For $j = 1, \ldots, m$
$\{\tilde{\mathbf{x}}_{j,t}^i\}_{i=1}^{N(j)} \sim p\left(\mathbf{x}_t|\{\mathbf{x}_{j,t-1}^i\}_{i=1}^{N(j)}\right)$, $\{\tilde{w}_{j,t}^i\}_{i=1}^{N(j)} = 1$
for k iterations **do**
Move $\{\tilde{\mathbf{x}}_{\cdot,t}^i\}_{i=1}^{N(j)}$, $\{\tilde{w}_{\cdot,t}^i\}_{i=1}^{N(j)}$ according to matrix $\mathcal{A}$
for $j = 1$ to m **do**
$\{\tilde{\mathbf{x}}_{j,t}^i\}_{i=1}^{N(j)} = \bigcup_{l\in N_j}\{\tilde{x}_{l,t}^i\}_{i\in\mathcal{I}_{l\to j}}$
$\{\tilde{w}_{j,t}^i\}_{i=1}^{N(j)} = \bigcup_{l\in N_j}\{\tilde{w}_{l,t}^i\}_{i\in\mathcal{I}_{l\to j}}$
$\{\tilde{w}_{j,t}^i\}_{i=1}^{N(j)} \leftarrow \{\tilde{w}_{j,t}^i\}_{i=1}^{N(j)} \times \rho_j(\mathbf{z}_{j,t}|\{\tilde{\mathbf{x}}_{j,t}^i\}_{i=1}^{N(j)})^{\frac{2|E(G)|}{kd(j)}}$
end for
end for
Resample: For $j = 1, \ldots, m$
Resample $\{\mathbf{x}_{j,t}^i\}_{i=1}^{N(j)}$ according to $\{\tilde{w}_{j,t}^i\}_{i=1}^{N(j)}$ and set weights $\{w_{j,t}^i\}_{i=1}^{N(j)} = \frac{1}{N(j)}$

4. NUMERICAL EXAMPLE

4.1 Flocking Model

We use a flocking model with a control law introduced in Tanner et al. (2003). The model has K vehicles which

move on the plane with the following dynamics.

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{x}_{1,t} \\ \vdots \\ \mathbf{x}_{K,t} \end{bmatrix}, \quad f(\mathbf{x}_t) = \mathbf{x}_t + \begin{bmatrix} f_1(\mathbf{x}_t) \\ \vdots \\ f_K(\mathbf{x}_t) \end{bmatrix} \Delta t \tag{35}$$

where

$$\mathbf{x}_{i,t} = \begin{bmatrix} x_{i,t} \\ y_{i,t} \\ \theta_{i,t} \\ v_{i,t} \end{bmatrix}, \quad f_i(\mathbf{x}_t) = \begin{bmatrix} v_{i,t}\cos\theta_{i,t} \\ v_{i,t}\sin\theta_{i,t} \\ w_{i,t} \\ a_{i,t} \end{bmatrix} \tag{36}$$

for $i = 1, \ldots, K$, where the state variables represent the position, orientation and translational speed of the vehicles. The variables $w_{i,t}$ and $a_{i,t}$ are control inputs and are given as

$$a_{i,t} = -\nabla_{x_{i,t}} V_{i,t} \cos\theta_{i,t} - \nabla_{y_{i,t}} V_{i,t} \sin\theta_{i,t} \tag{37}$$

$$w_{i,t} = -k \sum_{j \sim i} |v_{i,t}||v_{j,t}| \left(\theta_{i,t} - \theta_{j,t}\right) + \frac{\nabla_{x_{i,t}} V_{i,t} \sin\theta_{i,t} - \nabla_{y_{i,t}} V_{i,t} \cos\theta_{i,t}}{|v_{i,t}|} \tag{38}$$

where k is a control gain. The function $V_{i,t}$ is given as

$$V_{ij,t}(\mathbf{x}_t) = \frac{1}{\|r_{ij,t}\|^2} + \log \|r_{ij,t}\|^2 \tag{39}$$

$$V_{i,t}(\mathbf{x}_t) = \sum_{j \neq i} V_{ij,t}(\mathbf{x}_t) \tag{40}$$

where $\|r_{ij,t}\|^2 = (x_{i,t} - x_{j,t})^2 + (y_{i,t} - y_{j,t})^2$. The covariance matrix of the process noise Q_t is chosen randomly and is proportional to Δt^2. The range-only measurement model is used and thus $h_j(\mathbf{x}_t)$ is given as follows.

$$h_j(\mathbf{x}_t) = \begin{bmatrix} ((x_{1,t} - p_j)^2 + (y_{1,t} - p_j)^2)^{1/2} \\ \vdots \\ ((x_{K,t} - p_j)^2 + (y_{K,t} - p_j)^2)^{1/2} \end{bmatrix}. \tag{41}$$

The covariance of the measurement noise is proportional to the estimated distance and is given as

$$R_{j,t} = \text{diag} \begin{bmatrix} ((\hat{x}_{1,t} - p_j)^2 + (\hat{y}_{1,t} - p_j)^2)^{1/2} \\ \vdots \\ ((\hat{x}_{K,t} - p_j)^2 + (\hat{y}_{K,t} - p_j)^2)^{1/2} \end{bmatrix}. \tag{42}$$

In the sensor network, each node is connected to two neighbors, forming a square in figure 1.

4.2 Regularizing the EKF and DEKF

It is known that the EKF may diverge for several reasons, including a poor linear approximation and very different measurement noise levels (Perea et al., 2007). Divergence of the estimator is defined to be when the RMSE defined in (43) is larger than the RMSE for the null estimator that always returns the origin as the state estimate. When divergence occurs, the information matrices $Y_{t|t-1}$ and $Y_{t|t}$ become ill-conditioned after the prediction or measurement update steps. To avoid divergence we use two methods of regularization. First, the system equations are modified so that the denominators in (38) and (39) are $|v_{i,t}| + \epsilon$ and $\|r_{ij,t}\|^2 + \epsilon$, respectively, for a small value ϵ. Second, if the condition number of the information matrix exceeds 10^6 then we constrain its singular values to lie within $[1, 10^6]$ by taking an SVD and capping values outside this range. These values were chosen heuristically based on cases where the estimation result was fairly good.

Table 1. Algorithm performance: RMSE values and the fraction of divergent runs (averaged over 1000 Monte Carlo runs)

Algorithm (4 vehicles, $\Delta t = 0.1$)	RMSE	Divergence
EKF	N/A	49.7%
DEKF ($k_{\text{con}} = 2$)	N/A	62.3%
Regularized EKF	1.1986	2.1%
Regularized DEKF ($k_{\text{con}} = 2$)	2.5427	2.3%
CPF ($N = 100$)	0.2871	0%
MCDPF ($N = 100, k_{\text{mc}} = 10$)	0.8011	0%

Table 2. Algorithm performance: RMSE values and the fraction of divergent runs (averaged over 1000 Monte Carlo runs)

Algorithm (4 vehicles, $\Delta t = 0.05$)	RMSE	Divergence
EKF	0.1236	0%
DEKF ($k_{\text{con}} = 2$)	0.1733	0.1%
Regularized EKF	0.1164	0%
Regularized DEKF ($k_{\text{con}} = 2$)	0.1579	0%
CPF ($N = 100$)	0.1921	0%
MCDPF ($N = 100, k_{\text{mc}} = 10$)	0.7453	0%

4.3 Performance Comparison

The estimator tracks the position of 4 vehicles with 4 sensor nodes with range-only measurements. The number of particles is 100 for both the CPF and MCDPF in total, which means that each node maintains 25 particles on average for the MCDPF. The number of Markov chain iterations is $k_{\text{mc}} = 10$ and the number of consensus iterations is $k_{\text{con}} = 2$. The estimation results are shown in fig. 1. The time step is $\Delta t = 0.1$ with one measurement per timestep. The performance of the different state estimations is evaluated by comparing the root mean squared error (RMSE) of the estimation result,

$$\text{RMSE} = \frac{\Delta t}{TK} \sum_{t=1}^{T/\Delta t} \sum_{i=1}^{K} \sqrt{(x_{i,t} - \hat{x}_{i,t|t})^2 + (y_{i,t} - \hat{y}_{i,t|t})^2} \tag{43}$$

where the total simulation time is $T = 4$.

The comparison of results is given in table 1. Note that the RMSE of the EKF and DEKF is noted as N/A because of the large fraction of divergent runs. By using regularization we can see that the divergence is suppressed but we still see a large RMSE. The position error at each time during the simulation is given in fig. 2. We observe that the EKF and DEKF quickly diverge at around $t = 2.2$ while the regularized EKF (REKF) and regularized DEKF (RDEKF) do not diverge.

4.4 Effect of Δt

As we decrease Δt, we expect the error of EKF and DEKF to become negligible and the performance of EKF to beat that of CPF for very small Δt since the nonlinear system is well approximated by linearization. With the same numerical example using a smaller $\Delta t = 0.05$ the same performance comparison is shown in table 2. With the faster measurement rate, the RMSE for the EKF, DEKF, REKF and RDEKF decreases much more significantly than that of the CPF and MCDPF. Furthermore it becomes very unlikely that we have a divergent run. This illustrates the fact that for sufficiently small Δt the

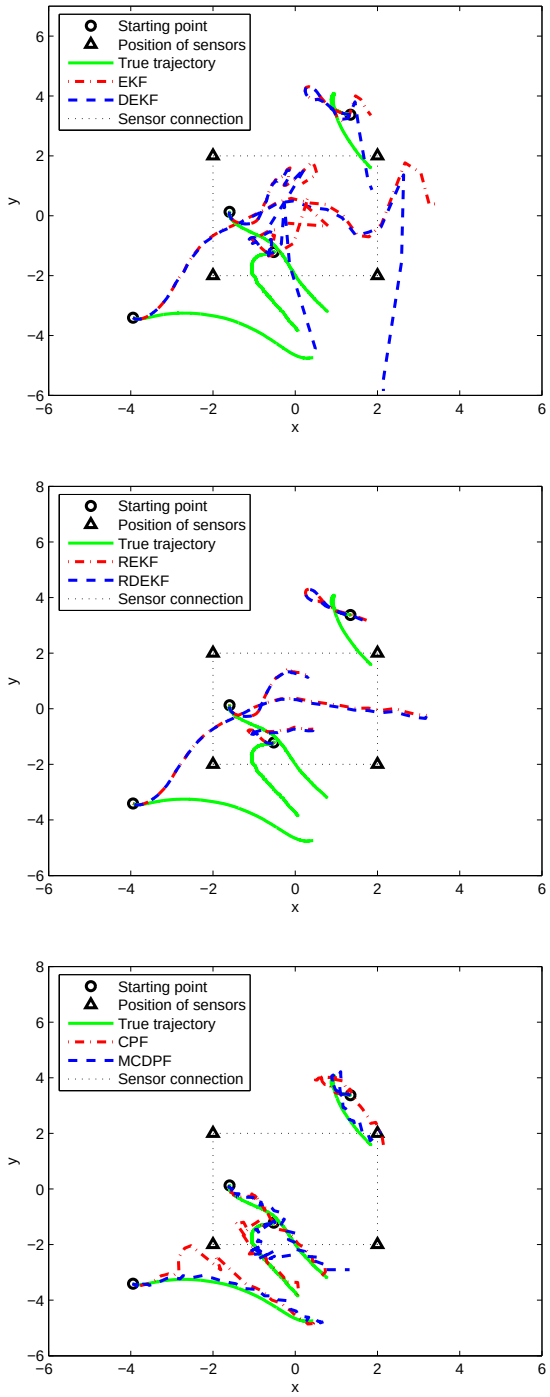

Fig. 1. Trajectory of the flocking model and its position estimation by the EKF, DEKF, REKF, RDEKF, CPF and MCDPF (top to bottom).

Kalman filter based approaches are superior, whereas for large Δt with significant nonlinearities the particle filters are better. This is due to the fact that the accuracy of the linearization varies inversely with Δt.

4.5 Effect of System Complexity

We now change the number of vehicle to 10, so the dimension of the system is increased from 16 to 40. The same timesteps, $\Delta t = 0.1$ and $\Delta t = 0.05$, are used and the RMSE is shown in table 3.

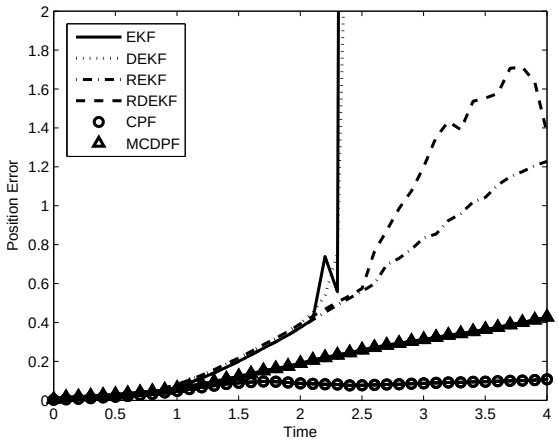

Fig. 2. RMSE versus time for the EKF, DEKF, REKF, RDEKF, CPF and MCDPF

Table 3. Algorithm performance: RMSE values and the fraction of divergent runs (averaged over 1000 Monte Carlo runs)

Algorithm (10 vehicles, $\Delta t = 0.1$)	RMSE	Divergence
EKF	N/A	100%
DEKF ($k_{\text{con}} = 2$)	N/A	100%
Regularized EKF	1.7108	8%
Regularized DEKF ($k_{\text{con}} = 2$)	12.7356	16.7%
CPF ($N = 100$)	0.4987	0%
MCDPF ($N = 100, k_{\text{mc}} = 10$)	0.7416	0%
Algorithm (10 vehicles, $\Delta t = 0.05$)	**RMSE**	**Divergence**
EKF	N/A	100%
DEKF ($k_{\text{con}} = 2$)	N/A	100%
Regularized EKF	1.5674	2.9%
Regularized DEKF ($k_{\text{con}} = 2$)	5.8275	6.1%
CPF ($N = 100$)	0.2434	0%
MCDPF ($N = 100, k_{\text{mc}} = 10$)	0.4807	0%

We see that increased system complexity dramatically reduces the performance of the Kalman filter based approaches, which no longer give acceptable performance even for the smaller timestep. In contrast, the particle filters (including the MCDPF) are robust with respect to the increase in system complexity, with very little loss of accuracy. We expect that much smaller Δt would eventually stabilize the EKF, DEKF and improve the RMSE, but for sufficiently large numbers of vehicles this may be prohibitively expensive.

4.6 Information Exchange Bandwidth

It is worthwhile to quantify the bandwidth (amount of data to be sent per unit time) at each node for the purpose of practical design of estimators for sensor networks. The information is communicated between connected sensors within the time interval of each new measurement, Δt. Suppose we have a fully connected sensor network with m sensors, N particles for the MCDPF, n system dimensions, k_{con} consensus iteration steps, k_{mc} Markov chain iteration steps, and Δt timestep for measurements. Taking a single scalar number as the basic unit of measurement (ignoring quantization issues), then BW (bandwidth, measured in

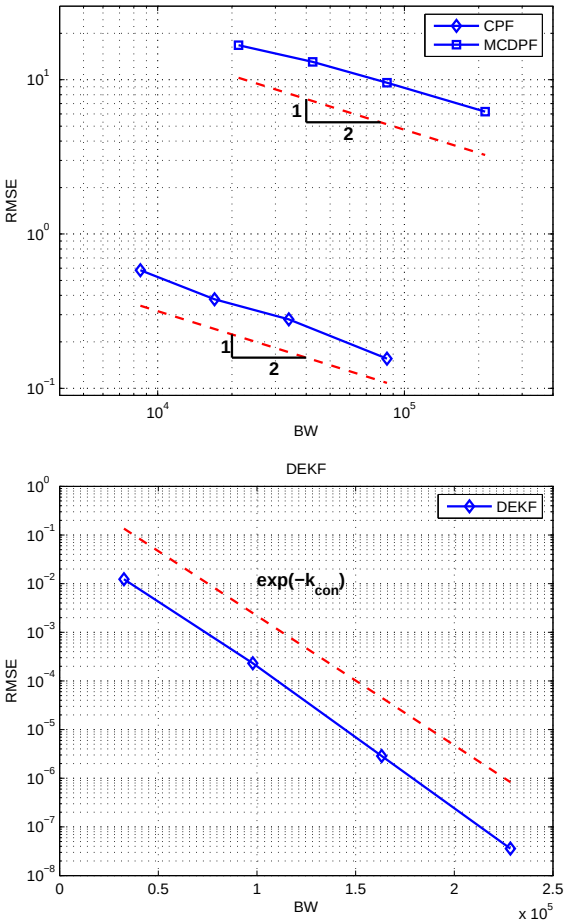

Fig. 3. RMSE with respect to BW with changing $N = 50, 100, 200, 500$ (CPF), $k_{\text{mc}} = 5, 10, 20, 50$ (MCDPF) and $k_{\text{con}} = 2, 6, 10, 14$ (DEKF). The decrease in RMSE is observed with increased BW.

numbers per unit time) for the DEKF, MCDPF and CPF are computed as follows.

$$\text{BW}_{\text{DEKF}} = \frac{k_{\text{con}}(n(n+1)(m-1))}{\Delta t} \tag{44}$$

$$\text{BW}_{\text{MCDPF}} = \frac{k_{\text{mc}}\left(\frac{N}{m}(n+1)\right)}{\Delta t} \tag{45}$$

$$\text{BW}_{\text{CPF}} = \frac{N(n+1)}{\Delta t}. \tag{46}$$

For a given system and sensor network, n, m and Δt are fixed and the parameters we can choose are k_{con}, k_{mc} and N. It is known that consensus schemes have a exponential convergence rate in term of k_{con} and the MCDPF has a geometric convergence rate in k_{mc} and N, as seen in figure 3, allowing bandwidth constrained filters to be designed.

5. CONCLUSION

Along with the review of the DEKF and MCDPF estimators, we described the nominal EKF and a regularization strategy, as well as comparing their performance numerically for a vehicle flocking model with 4 and 10 vehicles. We showed that the DEKF had lower RMSE than the MCDPF for simple systems (fewer vehicles) or high measurement frequency (small Δt), whereas for complex systems (more vehicles) or low measurement frequency (large Δt) we found that the MCDPF was both more robust and more accurate than the DEKF. In addition, the DEKF experienced divergence for complex or infrequently measured systems, whereas the MCDPF always gave stable estimates.

REFERENCES

Anderson, B. and Moore, J. (1979). *Optimal Filtering*. Prentice Hall, Englewood, NJ.

Coates, M. (2004). Distributed particle filters for sensor networks. In *Proc. 3rd workshop on Information Processing in Sensor Networks (IPSN)*, 99–107.

Crisan, D. and Doucet, A. (2002). A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on Signal Processing*, 50, 736–746.

Doucet, A., De Freitas, N., and Gordon, N. (2001). *Sequential Monte Carlo methods in practice*. Springer-Verlag.

Gu, D. (2007). Distributed particle filter for target tracking. In *2007 IEEE International Conference on Robotics and Automation*, 3856–3861.

Gu, D., Sun, J., Hu, Z., and Li, H. (2008). Consensus based distributed particle filter in sensor networks. In *International Conference on Information and Automation, 2008. ICIA 2008*, 302–307.

J. Chen, S.K. and Somani, A. (2006). Distributed fault detection of wireless sensor networks. In *DIWANS '06*, 65–71.

Lee, S.H. and West, M. (2009). Markov Chain Distributed Particle Filters (MCDPF). In *48th IEEE Conference on Decision and Control/28th Chinese Control Conference.*, 5496–5501.

Olfati-Saber, R. (2007). Distributed Kalman filtering for sensor networks. In *2007 46th IEEE Conference on Decision and Control*, 5492–5498.

Olfati-Saber, R. and Shamma, J.S. (2005). Consensus filters for sensor networks and distributed sensor fusion. In *Proc. IEEE Conference on Decision and Control, European Control Conference*, 6698–6703.

Perea, L., How, J., Breger, L., and Elosegui, P. (2007). Nonlinearities in sensor fusion. Divergence issues in EKF, modified truncated SOF, and UKF. In *AIAA Guidance Navigation and Control Conference and Exhibit*. Hilton Head, South Carolina.

Sheng, X., Hu, Y., and Ramanathan, P. (2005). Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, 24. Piscataway, NJ, USA.

Tanner, H., Jadbabaie, A., and Pappas, G.J. (2003). Coordination of multiple autonomous vehicles. In *IEEE Mediterranean Conference on Control and Automation*.

Xiao, L., Boyd, S., and Lall, S. (2006). A space-time diffusion scheme for peer-to-peer least-squares estimation. In *IPSN '06: Proceedings of the 5th international conference on Information processing in sensor networks*, 168–176.

Yan, T., He, T., and Stankovic, J.A. (2003). Differentiated surveillance for sensor networks. In *Proc. ACM Int'l Conf. Embedded Networked Sensor Systems (SenSys '03)*, 51–62.