Decentralized Sigma-Point Information Filters for Target Tracking in Collaborative Sensor Networks

Tom Vercauteren, *Student Member, IEEE,* and Xiaodong Wang, *Senior Member, IEEE*

***Abstract*—Tracking a target in a cluttered environment is a representative application of sensor networks and a benchmark for collaborative signal processing algorithms. This paper presents a strictly decentralized approach to Bayesian filtering that is well fit for *in-network* signal processing. By combining the sigma-point filter methodology and the information filter framework, a class of algorithms denoted as *sigma-point information filters* is developed. These techniques exhibit the robustness and accuracy of the sigma-point filters for nonlinear dynamic inference while being as easily decentralized as the information filters. Furthermore, the computational cost of this approach is equivalent to a local Kalman filter running in each active node while the communication burden can be made linearly growing in the number of sensors involved. The proposed algorithms are then adapted to the specific problem of target tracking with data association ambiguity. Making use of a local probabilistic data association, we formulate a decentralized tracking scheme that significantly outperforms the existing schemes with similar computational and communication complexity.**

***Index Terms*—Decentralized filtering, information filter, sensor networks, sigma-point Kalman filter, target tracking.**

I. Introduction

DENSELY scattered low-cost sensor nodes provides a rich and complex information source about the sensed world. The convergence of recent developments in micro electro-mechanical systems (MEMS), microprocessors, and *ad hoc* networking protocols have made such sensor networks a not-so-far reality [1]. The randomly deployed sensor nodes of the network are able to collect and process data and communicate. The potential tasks achieved by sensor networks are far reaching and include event detection, event identification, and location sensing [2]. In order to make these functions feasible, it is necessary to develop specific collaborative algorithms taking into account the constraints of the low-cost nodes, such as limited computational capability and power resources.

We consider the problem of online inference in sensor networks. In [2] and [3], this problem is treated by selecting at each time a single sensor (the leader-node), which is responsible for the data acquisition and data fusion. Such a scheme seems well fit for our concerns because it is power efficient. However, typical sensor nodes are usually prone to errors, and this algorithm would suffer dramatically from the loss of a sensor. Moreover, the issue of multisensor fusion was not really addressed since it was solved by activating several sensors and let them send their measurements to the leader node.

The goal of this work is to develop a fully decentralized scheme for online inference so that at each time, the set of activated sensors collects a diverse information and efficiently combines it. A fully decentralized system is defined in [4] as a structure in which all information is processed locally, and no central processing site arises. The advantage of such a scheme is multifold. First, the failure of a sensor does not imply the failure of the entire system. Second, the use of several sensors provides a resource of diversity in terms of quality of information (due to redundancy) as well as type of information if different kinds of sensors are used. An erroneous measurement does not necessarily imply a drastic error on the global estimate.

A framework for decentralized estimation can be found in [4]. The basis of this framework relies on an algebraic equivalence of the well-known Kalman filter expressed in terms of measures of information, namely, the *information filter* [4]–[7]. This scheme provides optimal estimation when applied to linear-Gaussian systems and was extended to the nonlinear case through a similar procedure as the one used to derive the extended Kalman filter (EKF) [4], [7]. This scheme suffers from the same flaws as the EKF sometimes leading to the divergence of the inference system. Recently, a number of related approaches that are more accurate and more robust have emerged. This includes the *unscented Kalman filter* (UKF) [8], [9] and the *central difference Kalman filter* (CDKF) [10]. These algorithms are based on derivativeless statistical linearization from a choice of *sigma-points* and were unified in [11] and [12]. In our work, this methodology is used to derive a class of fully decentralized filters called *sigma-point information filters* (SPIFs) that inherits from the advantages of these algorithms.

The effectiveness of the proposed algorithms for sensor networks applications resides in the fully decentralized facet discussed above and in their communication and computation efficiency. The computational complexity is equivalent to a local Kalman filter running at each node, and the number of communications can be made linear in terms of the number of sensors. The proposed scheme is thus more power efficient than the decentralized particle filter solutions of [13] and [14]. The advantage of the particle filters [15], [16] is that no parametric form of the posterior is assumed, and this allows for more general solutions. If the same flexibility cannot be achieved by our algorithm, it can still cope with multimodal densities through a proposed Gaussian-mixture extension. The loss of information

Manuscript received May 11, 2004; revised October 25, 2004. This work was supported in part by the U.S. National Science Foundation under Grants DMS-0244583 and CCR-0207550 and in part by the U.S. Office of Naval Research (ONR) under Grant N00014-03-1-0039. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dominic K. C. Ho.

T. Vercauteren was with the Department of Electrical Engineering, Columbia University, New York, NY 10027 USA. He is now with INRIA, Sophia-Antipolis, France.

X. Wang is with the Department of Electrical Engineering, Columbia University, New York, NY 10027 USA (e-mail: wangx@ee.columbia.edu).

Digital Object Identifier 10.1109/TSP.2005.851106

induced by the Gaussian assumption of the Kalman-based filters can be significant. It is, however, known [17] that any probability density can be approximated with arbitrary precision by a Gaussian mixture. Therefore, Gaussian mixtures have been widely used for non-Gaussian filtering [10], [12], [17]–[19]. In those works, the algorithms rely on a bank of parallel Gaussian-based filters. This idea can be included in the information filter framework to get the *decentralized Gaussian mixture sigma-point information filters* (DGMSPIFs).

We further illustrate the usefulness of the sigma-point information filters for in-network processing by applying these techniques to the problem of collaborative target tracking in sensor networks. The tracking problem differs from the usual nonlinear dynamic state-space because at each time step, each sensor acquires a set of measurements with at most one observation corresponding to the target of interest. This ambiguity is known as the data association problem and is usually solved by using the most probable measurements of the set, which is known as the nearest neighbor (NN) rule, or by a probabilistic data association (PDA) rule [20]. The NN scheme is known to diverge easily and the extension of the PDA to the multisensor scenario [21] has a prohibitive complexity. Consequently, we develop a local probabilistic data association (LPDA) scheme in order to keep the same complexity as the NN method while avoiding the divergence of the estimates.

The remainder of this paper is organized as follows. Section II presents the proposed sigma-point information filters for decentralized inference. In Section III, we adapt the proposed algorithms to the problem of target tracking in collaborative sensor networks. Simulation results are presented in Sections IV, and Section V concludes the paper.

II. Decentralized Inference in Dynamic Systems

We consider a generic discrete-time dynamic state-space model described by

$$\text{initial state model } p(x(0)) \tag{1}$$

$$\text{state transition model } x(t) = f_t[x(t-1), v(t)] \tag{2}$$

$$\text{observation model } z(t) = h_t[x(t), w(t)] \tag{3}$$

where $x(t)$ is the state vector, $z(t)$ is the observation vector, $v(t)$ represents the process noise, and $w(t)$ denotes the observation noise. It is further assumed that the noise vectors $v(t)$ and $w(t)$ are white, zero-mean, and uncorrelated: $E[v(t)v^T(t')] = \delta(t-t')Q(t)$, $E[w(t)w^T(t')] = \delta(t-t')R(t)$ and $E[v(t)w^T(t')] = 0$. The cumulative set of measurements is denoted by $Z^t \triangleq \{z(1), \ldots, z(t)\}$. Suppose an online inference of $x(t)$ is of interest. That is, at current time t, we wish to make an estimate of a function of the state variable $x(t)$, say $\psi[x(t)]$, based on the currently available observations Z^t. In such a Bayesian framework, the posterior filtering density $p(x(t) \mid Z^t)$ provides the complete solution to the probabilistic inference problem under consideration. The filtering density can be recursively updated by the following sequential algorithm:

PREDICTION

$$p(x(t) \mid Z^{t-1}) = \int p(x(t) \mid x(t-1)) p(x(t-1) \mid Z^{t-1}) \, \mathrm{d}x(t-1) \tag{4}$$

UPDATE

$$p(x(t) \mid Z^t) \propto p(z(t) \mid x(t)) p(x(t) \mid Z^{t-1}). \tag{5}$$

A. Decentralized Information Filter

For linear, Gaussian systems, the optimal closed-form solution to (4) and (5) is known as the Kalman filter [22]. The information filter (which is also called inverse covariance form of the Kalman filter) [4]–[7] is an algebraic equivalence of the Kalman filter. It is expressed in terms of measures of information about the state rather than the usual state estimate and covariance. The motivation for introducing this algorithm lies in its capability of being easily decentralized.

We consider a linear and Gaussian system

$$\text{initial state model} \quad x(0) \sim \mathcal{N}(\hat{x}(0 \mid 0), P(0 \mid 0)) \tag{6}$$

$$\text{state transition model} \quad x(t) = F(t)x(t-1) + G(t)v(t) \tag{7}$$

$$\text{observation model} \quad z(t) = H(t)x(t) + w(t) \tag{8}$$

where $\hat{x}(t \mid t') \triangleq E[x(t) \mid Z^{t'}]$, and $P(t \mid t') \triangleq \mathrm{Var}(x(t) \mid Z^{t'})$. The Kalman filter provides an expression of the prediction and update steps (4) and (5) in terms of mean and variance of the variable [22]. The information filter is derived from it by expressing the recursion in terms of the information matrix Y and the information state vector $\hat{y}$

$$Y(t \mid t') \triangleq P^{-1}(t \mid t') \tag{9}$$

$$\hat{y}(t \mid t') \triangleq P^{-1}(t \mid t')\hat{x}(t \mid t') = Y(t \mid t')\hat{x}(t \mid t'). \tag{10}$$

The Kalman filter can be re-expressed as

PREDICTION

$$\hat{y}(t \mid t-1) = L(t)\hat{y}(t-1 \mid t-1) \tag{11}$$

$$Y(t \mid t-1) = [F(t)Y^{-1}(t-1 \mid t-1)F(t)^T + G(t)Q(t)G^T(t)]^{-1} \tag{12}$$

UPDATE

$$\hat{y}(t \mid t) = \hat{y}(t \mid t-1) + \underbrace{H^T(t)R^{-1}(t)z(t)}_{i(t)} \tag{13}$$

$$Y(t \mid t) = Y(t \mid t-1) + \underbrace{H^T(t)R^{-1}(t)H(t)}_{I(t)} \tag{14}$$

where the information propagation coefficient is given by $L(t) = Y(t \mid t-1)F(t)Y^{-1}(t-1 \mid t-1)$, and $i(t), I(t)$ are the information contributions. The derivations can be found in many textbooks such as [4]–[7].

The above information filter is used to provide a decentralized inference algorithm [4], [7], [21], [23]. The available observations consist of measurements taken at S different sensors. The observation model is divided into S equations

$$z_s(t) = H_s(t)x(t) + w_s(t), s = 1 \ldots S \tag{15}$$

where the observation noise is uncorrelated and Gaussian, i.e.,

$$E\left[w_s(t)w_{s'}^T(t')\right] = \delta(t-t')\delta(s-s')R_s(t). \tag{16}$$

The equivalent centralized system is given by the aggregated matrices

$$\begin{aligned} H(t) &= \left[H_1^T(t), \ldots, H_S^T(t)\right]^T \\ v(t) &= \left[v_1^T(t), \ldots, v_S^T(t)\right]^T \\ R(t) &= \text{diag}\{R_1(t), \ldots, R_S(t)\}. \end{aligned} \tag{17}$$

Using the particular form of the observation noise covariance, the information contributions $i(t), I(t)$ can be expanded as

$$\begin{aligned} i(t) &= H^T(t)R^{-1}(t)z(t) \\ &= \sum_{s=1}^{S} \underbrace{H_s^T(t)R_s^{-1}(t)z_s(t)}_{i_s(t)} \end{aligned} \tag{18}$$

$$\begin{aligned} I(t) &= H^T(t)R^{-1}(t)H(t) \\ &= \sum_{s=1}^{S} \underbrace{H_s^T(t)R_s^{-1}(t)H_s(t)}_{I_s(t)} \end{aligned} \tag{19}$$

where the local information contributions $i_s(t)$ and $I_s(t)$ only require local knowledge and computations at sensor s. The information contributions of the entire set of observations are simply the sum of the local contributions. At each time step, the sensors need to communicate so that the sum of the local information contributions are available to each sensor. This could be done by using a communication chain, a tree structured network, or a broadcast medium. If a communication chain is used, a two-pass communication link spreads the quantity with a number of communications growing linearly with the number of sensors. The information filter is thus a natural form for decentralizing the Kalman filter.

B. Statistical Linear Regression

Most real-world systems are nonlinear and non-Gaussian, and in this case, the optimal framework above cannot be straightforwardly used. Approximate solutions must then be employed. To extend the information filtering to nonlinear systems, we will first make the assumption that the state and the noise terms can be represented by a Gaussian distribution. The filtering distribution is then simply specified by its mean and covariance and the inference problem (4) and (5) can be simplified into

PREDICTION

$$\hat{x}(t \,|\, t-1) = E\left[f_t[x(t-1), v(t)] \,|\, Z^{t-1}\right] \tag{20}$$

$$P(t \,|\, t-1) = \text{Var}[f_t[x(t-1), v(t)] \,|\, Z^{t-1}] \tag{21}$$

UPDATE

$$\hat{x}(t \,|\, t) = \hat{x}(t \,|\, t-1) + W(t)[z(t) - \hat{z}(t \,|\, t-1)] \tag{22}$$

$$P(t \,|\, t) = P(t \,|\, t-1) - W(t)S(t)W^T(t) \tag{23}$$

where the predicted measurement mean is given by $\hat{z}(t \,|\, t-1) = E[h_t[x(t), w(t)] \,|\, Z^{t-1}]$, its variance is $S(t) = \text{Var}(z(t) \,|\, Z^{t-1})$ and $W(t) = \text{Cov}(x(t), z(t) \,|\, Z^{t-1})S^{-1}(t)$ is the so called gain matrix.

In order to approximate the quantities of interest here, the EKF linearizes (2) and (3) around a single point ($\hat{x}(t-1 \,|\, t-1)$ for the prediction and $\hat{x}(t \,|\, t-1)$ for the observation update). This solution does not account for the spread of the random variables and uses only the first-order Taylor expansion of the nonlinear functions. Therefore, it often leads to the divergence of the filter. To avoid the flaws of the EKF, several recent works [8]–[12] (referred hereafter to as *sigma-point Kalman filters*) have used deterministic sampling techniques and the sigma-point transform to propagate the mean and variance.

a) Sigma-Point Transform: Let $g(\cdot)$ be some nonlinear function and x be some random variable with mean $\hat{x}$ and variance P_{xx}. We wish to estimate the mean and variance of $u = g(x)$. The sigma-point transform uses a set of weighted points $\{\mathcal{X}_i, \omega_i\}_{i=1\ldots r}$ called sigma-points to represent the p.d.f. of x. It is chosen such that the mean and covariance are consistent with the prior information $\bar{x} = \hat{x}$ and $\bar{P}_{xx} = P_{xx}$, with $\bar{x} \triangleq \sum_{i=1}^{r} \omega_i \mathcal{X}_i$, $\bar{P}_{xx} \triangleq \sum_{i=1}^{r} \omega_i (\mathcal{X}_i - \bar{x})(\mathcal{X}_i - \bar{x})^T$, and $\sum_{i=1}^{r} \omega_i = 1$. The sigma-points are then propagated through the nonlinear function $\mathcal{U}_i = g(\mathcal{X}_i)$ to approximate the mean and covariance of u:

$$E[u] \approx \bar{u} \triangleq \sum_{i=1}^{r} \omega_i \mathcal{U}_i \tag{24}$$

$$\text{Var}(u) \approx \bar{P}_{uu} \triangleq \sum_{i=1}^{r} \omega_i (\mathcal{U}_i - \bar{u})(\mathcal{U}_i - \bar{u})^T \tag{25}$$

$$\text{Cov}(x, u) \approx \bar{P}_{xu} \triangleq \sum_{i=1}^{r} \omega_i (\mathcal{X}_i - \bar{x})(\mathcal{U}_i - \bar{u})^T. \tag{26}$$

The sigma-point Kalman filters approximate the Gaussian inference (20)–(23) by using this algorithm to propagate means and covariances. Within this framework, the different available algorithms differ in the way they specify the initial set of weighted sigma-points so as to capture the most important information about the random variable of interest. Besides of the consistency with the prior information, it is possible to specify some additional constraints on the set of *sigma-points*, the choice of the constraints make each algorithm specific. See [9], [10], and [12] for detailed account of the choices. We will only summarize the unscented Kalman filter (UKF) for illustration purposes. In this algorithm, the n-dimensional random variable x is approximated by $2n+1$ sigma-points given by

$$\begin{aligned} \mathcal{X}_0 &= \hat{x}, \quad \omega_0 = \frac{\kappa}{n+\kappa} \\ \mathcal{X}_i &= \hat{x} + (\sqrt{(n+\kappa)P_{xx}})_i, \quad \omega_i = \frac{1}{2(n+\kappa)} \\ &\qquad\qquad\qquad\qquad i = 1 \ldots n \\ \mathcal{X}_{i+n} &= \hat{x} - (\sqrt{(n+\kappa)P_{xx}})_i, \quad \omega_{i+n} = \frac{1}{2(n+\kappa)} \\ &\qquad\qquad\qquad\qquad i = 1 \ldots n \end{aligned} \tag{27}$$

where κ is a scaling parameter usually chosen as 0 or $3-n$ and $(\sqrt{P})_i$ denotes the ith row of the Cholesky decomposition of P. It is shown in [8] that this procedures produces accurate results for the predicted mean and covariance up to the third order of the Taylor series for Gaussian inputs and at least up to the second order for other types of inputs.

It has been shown in [11] and [12] that the the sigma-point transform can also be seen as a statistical linearization of $g(\cdot)$. The statistical linear regression computes A and c such that $u =$

$g(x) \approx Ax + c$ and therefore seeks to minimize the weighted sum of squared errors $e_i = \mathcal{U}_i - (A\mathcal{X}_i + c)$:

$$\{A, c\} = \arg\min_{A,c} \sum_{i=1}^{r} \omega_i e_i^T e_i. \tag{28}$$

The solution to (28) is the usual least square fitting: $A = \bar{P}_{xu}^T \bar{P}_{xx}^{-1}$ and $c = \bar{u} - A\bar{x}$. The mean and covariance matrix of the deviations e_i is then

$$\bar{e} = \sum_{i=1}^{r} \omega_i e_i = \bar{u} - A\bar{x} - c = 0 \tag{29}$$

$$\bar{P}_{ee} = \sum_{i=1}^{r} \omega_i e_i e_i^T = \bar{P}_{uu} - A\bar{P}_{xx}A^T. \tag{30}$$

The propagation of the mean and covariances through the linearization

$$g^{\text{lin}}(x) = Ax + c + e \tag{31}$$

where e is assumed to be zero-mean with covariance matrix $\bar{P}_{ee}$ and uncorrelated with x, provides the same approximations as (24)–(26):

$$\begin{aligned} E[g^{\text{lin}}(x)] &= A\bar{x} + (\bar{u} - A\bar{x}) + 0 = \bar{u} \\ \text{Var}(g^{\text{lin}}(x)) &= A\bar{P}_{xx}A^T + (\bar{P}_{uu} - A\bar{P}_{xx}A^T) = \bar{P}_{uu} \\ \text{Cov}(x, g^{\text{lin}}(x)) &= \text{Var}(x)A^T = \bar{P}_{xx}\left[\bar{P}_{xu}^T \bar{P}_{xx}^{-1}\right]^T = \bar{P}_{xu}. \end{aligned}$$

b) Linearized State-Space: From the previous linearization of a function we can now extract a statistical linearization of the dynamic state-space. The idea is to use augmented vectors composed of the state and noise terms and propagate them through augmented functions so that the updates correspond to propagations through nonlinear functions with one vector input.

We denote $x^{ab}(t-1) \triangleq [x^T(t-1), v^T(t), w^T(t)]^T$, $x^a(t-1) \triangleq [x^T(t-1), v^T(t)]^T$, and $x^b(t) \triangleq [x^T(t), w^T(t)]^T$ as the augmented states. Note that x^b is indexed by t and not $t-1$ as opposed to x^{ab} and x^a. At time t, our prior information is summarized by

$$x^{ab}(t-1) \sim \mathcal{N}(\underbrace{[\hat{x}(t-1\,|\,t-1)^T, 0, 0]^T}_{\bar{x}^{ab}(t-1\,|\,t-1)}, \underbrace{\text{diag}\{P(t-1\,|\,t-1), Q(t), R(t)\}}_{\bar{P}^{ab}(t-1\,|\,t-1)}). \tag{32}$$

Similarly we have $\bar{x}^a(t-1) = [\hat{x}(t-1\,|\,t-1)^T, 0]^T$ and $\bar{P}^a(t-1) = \text{diag}\{P(t-1\,|\,t-1), Q(t)\}$. Once the predicted state statistics are given we also assume that

$$x^b(t-1) \sim \mathcal{N}(\underbrace{[\hat{x}(t\,|\,t-1)^T, 0)}_{\bar{x}^b(t\,|\,t-1)}; \underbrace{\text{diag}\{\bar{P}(t\,|\,t-1), R(t)\}}_{\bar{P}^b(t\,|\,t-1)}). \tag{33}$$

The state-space model (2)–(3) can be rewritten as a set of two nonlinear functions of these augmented states

$$x(t) = f_t^a[x^a(t-1)] \tag{34}$$

$$z(t) = h_t^b[x^b(t)]. \tag{35}$$

From (31), the state-space (2)–(3) can be linearized into

$$x(t) = \bar{F}(t)x(t-1) + c^x(t) + \bar{v}(t) \tag{36}$$

$$z(t) = \bar{H}(t)x(t) + c^z(t) + \bar{w}(t) \tag{37}$$

where we have separated the components corresponding to the state and the noise.

To obtain this linearization, a set of augmented sigma-points $\{\mathcal{X}_i^{ab}(t-1\,|\,t-1), \omega_i\}$ is generated by (27) to describe $x^{ab}(t-1)$. The part of those points corresponding to $x^a(t-1)$ is denoted by $\mathcal{X}_i^a(t-1\,|\,t-1)$. Each of those sigma-points is processed through the transition equation $\mathcal{X}_i(t\,|\,t-1) \triangleq f_t^a[\mathcal{X}_i^a(t-1\,|\,t-1)]$. The information about $w(t)$ is captured by the component $\mathcal{W}_i(t)$ of $\mathcal{X}_i^{ab}(t-1\,|\,t-1)$ corresponding to $w(t)$ and is used to create a set of augmented sigma-points $\mathcal{X}_i^b(t\,|\,t-1) \triangleq [\mathcal{X}_i^T(t\,|\,t-1), \mathcal{W}_i^T(t)]^T$ which is propagated through (35) to get the sigma-points $\mathcal{Z}_i^b(t\,|\,t-1) \triangleq h_t^b[\mathcal{X}_i^b(t\,|\,t-1)]$ corresponding to the observations. The state transition is linearized from (31) as

$$\begin{aligned} x(t) &= [\bar{F}(t), \bar{G}(t)]x^a(t-1) + c^x(t) + e_f(t) \\ &= \bar{F}(t)x(t-1) + \bar{G}(t)v(t) + c^x(t) + e_f(t) \end{aligned}$$

where

$$\begin{aligned} [\bar{F}(t), \bar{G}(t)] &= \bar{P}^a(t, t-1\,|\,t-1)\bar{P^a}^{-1}(t-1\,|\,t-1) \\ c^x(t) &= \bar{x}(t\,|\,t-1) - [\bar{F}(t), \bar{G}(t)]\bar{x}^a(t-1\,|\,t-1) \\ &= \bar{x}(t\,|\,t-1) - \bar{F}(t)\bar{x}(t-1\,|\,t-1). \end{aligned} \tag{38}$$

The linearization can thus be rewritten as (36) by defining the total process noise

$$\bar{v}(t) = \bar{G}(t)v(t) + e_f(t) \tag{39}$$

as the sum of the linearized actual process noise $\bar{G}(t)v(t)$ and the linearization noise $e_f(t)$. The variance of this total process noise is given by

$$\begin{aligned} \bar{Q}(t) &= \bar{G}(t)Q(t)\bar{G}^T(t) + \bar{P}_{e_f}(t) \\ &= \bar{G}(t)Q(t)\bar{G}^T(t) + \bar{P}(t\,|\,t-1) \\ &\quad - [\bar{F}(t), \bar{G}(t)]\bar{P}^a(t-1\,|\,t-1)[\bar{F}(t), \bar{G}(t)]^T \\ &= \bar{P}(t\,|\,t-1) - \bar{F}(t)\bar{P}(t-1\,|\,t-1)\bar{F}^T(t). \end{aligned} \tag{40}$$

The observation model is linearized from (31) as

$$\begin{aligned} z(t) &= [\bar{H}(t), \bar{K}(t)]x^b(t) + c^z(t) + e_h(t) \\ &= \bar{H}(t)x(t) + \bar{K}(t)w(t) + c^z(t) + e_h(t). \end{aligned}$$

where

$$\begin{aligned} [\bar{H}(t), \bar{K}(t)] &= \bar{P}_{zx}^b(t\,|\,t-1)\bar{P^b}^{-1}(t\,|\,t-1) \\ c^z(t) &= \bar{z}(t\,|\,t-1) - [\bar{H}(t), \bar{K}(t)]\bar{x}^b(t\,|\,t-1) \\ &= \bar{z}(t\,|\,t-1) - \bar{H}(t)\bar{x}(t\,|\,t-1). \end{aligned} \tag{41}$$

The linearization can thus be rewritten as (37) by defining the total observation noise

$$\bar{w}(t) = \bar{K}(t)w(t) + e_h(t) \tag{42}$$

as the sum of the linearized actual observation noise and the linearization noise. The variance of this total observation noise is given by

$$\begin{aligned}\bar{R}(t) &= \bar{K}(t)R(t)\bar{K}^T(t) + \bar{P}_{e_h}(t)\\ &= \bar{K}(t)R(t)\bar{K}^T(t) + \bar{S}(t)\\ &\quad - [\bar{H}(t), \bar{K}(t)]\bar{P}^b(t\,|\,t-1)[\bar{H}(t), \bar{K}(t)]^T\\ &= \bar{S}(t) - \bar{H}(t)\bar{P}(t\,|\,t-1)\bar{H}^T(t)\end{aligned} \tag{43}$$

where $\bar{S}(t) = \bar{P}_{zz}(t\,|\,t-1)$ is the approximated variance of the innovation.

If the observation noise is additive then the above equations can be simplified. It is not necessary to include $w(t)$ in the sigma-points. Furthermore if one of the equation of the state-space is linear (typically the state equation), then the statistics corresponding to this part can be explicitly computed to simplify the procedure.

C. Sigma-Point Information Filters

By applying the information filter (11)–(14) to the linearized state-space (36)–(37), the sigma-point information filters (SPIF) for robust decentralized inference in nonlinear systems are derived:

PREDICTION

$$\hat{y}(t\,|\,t-1) = Y(t\,|\,t-1)[\bar{F}(t)\hat{x}(t-1\,|\,t-1) + c^x(t)] \tag{44}$$

$$Y(t\,|\,t-1) = [\bar{F}(t)P(t-1\,|\,t-1)\bar{F}^T(t) + \bar{Q}(t)]^{-1} \tag{45}$$

UPDATE

$$\hat{y}(t\,|\,t) = \hat{y}(t\,|\,t-1) + \underbrace{\bar{H}^T(t)\bar{R}^{-1}(t)[z(t) - c^z(t)]}_{i(t)} \tag{46}$$

$$Y(t\,|\,t) = Y(t\,|\,t-1) + \underbrace{\bar{H}^T(t)\bar{R}^{-1}(t)H(t)}_{I(t)}. \tag{47}$$

On one hand, the sigma-point information filter is an information filter applied to the linear state-space obtained by statitical linearization. On the other hand, we have previously shown that the sigma-point Kalman filter is a usual Kalman filter running on the very same linarized state space. On this linear state-space, the information filter and the Kalman filter are strictly equivalent. Therefore, the sigma-point information filter is equivalent to a sigma-point Kalman filter.

Previously, we have shown how the information filter could be easily distributed. This conclusion derived from the particular form of the observation noise covariance, which was block diagonal. If we apply the statistical linearization concept to the aggregated observation model (17), we will introduce some nonzero cross terms in the total linearized observation noise covariance $\bar{R}$. Those cross terms appear because the spread of the state variables (common to all sensors) is taken into account to linearize the system and reflected in the linearization noise $e_h(t)$. This would prevent the decentralization of the information filter. Such cross-terms do not appear in the EKF because there, the effect of the linearization is never compensated.

In order to be able to decentralize the sigma-point information filter, we will then use a local linearization of the observation equations. The cross-terms of the global approach are likely to be small, and therefore, going to the local linearization should not cause a big discrepancy with the global linearization. In any case, the local sigma-point approach will outperform the EKF because the spread of the variable is taken into account. Linearizations of the local observation models are performed based on the previous algorithm, and we get S equations

$$z_s(t) = \bar{H}_s(t)x(t) + c_s^z(t) + \bar{w}_s(t), \forall s = 1 \ldots S \tag{48}$$

where $\bar{H}_s(t), c_s^z(t)$, and $\bar{w}_s(t)$ are computed as in (36)and (37). In this approximation, we assume that $\bar{R}(t) = \text{diag}\{\bar{R}_1(t), \ldots, \bar{R}_S(t)\}$, and thus

$$\begin{aligned}i(t) &= \bar{H}^T(t)\bar{R}^{-1}(t)[z(t) - c^z(t)]\\ &= \sum_{s=1}^{S} \underbrace{\bar{H}_s^T(t)\bar{R}_s^{-1}(t)[z_s(t) - c_s^z(s)]}_{i_s(t)}\end{aligned} \tag{49}$$

$$\begin{aligned}I(t) &= \bar{H}^T(t)\bar{R}^{-1}(t)\bar{H}(t)\\ &= \sum_{s=1}^{S} \underbrace{\bar{H}_s^T(t)\bar{R}_s^{-1}(t)\bar{H}_s(t)}_{I_s(t)}\end{aligned} \tag{50}$$

where the local information contributions $i_s(t)$ and $I_s(t)$ and the local total linearized observation noise covariance $\bar{R}_s(t)$ are computed as in (36) and (37). It can be seen that neither the process equation linearization nor the local observation equation linearization require the knowledge of the observation model of the other sensors. We also notice that since the linearization is deterministic, provided that all sensors are initialized with the same prior, they all compute the same prediction. Therefore, if they communicate at full rate, at each time step, they share the same belief.

c) Decentralized Gaussian Mixture Sigma-Point Information Filters: The system under consideration remains as presented previously, but we assume that at time $t-1$, the posterior distribution can be represented by a Gaussian mixture with M components to use a bank of parallel Gaussian-based filters. We assume that

$$\begin{aligned}p(x(t-1)\,|\,Z^{t-1}) = \sum_{m=1}^{M} &\underbrace{\alpha_m(t-1)}_{p(m\,|\,Z^{t-1})}\\ &\underbrace{\mathcal{N}(x(t-1); \hat{x}_m(t-1\,|\,t-1), P_m(t-1\,|\,t-1))}_{p(x(t-1)\,|\,Z^{t-1}, m)}\end{aligned} \tag{51}$$

where the random variable m means that the mth mixture was used. The prediction step becomes

$$\begin{aligned}&p(x(t)\,|\,Z^{t-1})\\ &\quad= \sum_{m=1}^{M} p(m\,|\,Z^{t-1})p(x(t)\,|\,Z^{t-1}, m)\\ &\quad\approx \sum_{m=1}^{M} \alpha_m(t-1)\mathcal{N}(x(t); \bar{x}_m(t\,|\,t-1), \bar{P}_m(t\,|\,t-1)).\end{aligned}$$

Our Gaussian approximation for $p(x(t)\,|\,Z^{t-1}, m)$ provides $\bar{x}_m(t\,|\,t-1)$ and $\bar{P}_m(t\,|\,t-1)$. We see that the prediction step

for the Gaussian mixture case corresponds to M parallel predictions, and the weights are kept identical. For the observation update, we find that

$$\begin{aligned} p(x(t)\,|\,Z^t) &= \sum_{m=1}^{M} p(m\,|\,Z^t)p(x(t)\,|\,Z^t,m) \\ &= \frac{1}{C}\sum_{m=1}^{M} p(z(t)\,|\,Z^{t-1},m)p(m\,|\,Z^{t-1})p(x(t)\,|\,Z^t,m) \\ &\approx \frac{1}{C}\sum_{m=1}^{M} \mathcal{N}(z(t);\bar{z}_m(t\,|\,t-1),\bar{S}_m(t)) \\ &\quad \times \alpha_m(t-1)p(x(t)\,|\,Z^t,m) \\ &= \sum_{m=1}^{M} \alpha_m(t)\mathcal{N}(x(t);\bar{x}_m(t\,|\,t),\bar{P}_m(t\,|\,t)) \end{aligned}$$

where C is a normalizing constant, and $S_m(t), \hat{x}_m(t\,|\,t)$, and $P_m(t\,|\,t)$ are provided by the Gaussian approximation applied to the mth mixture. The weights are updated through

$$\alpha_m(t) = \alpha_m(t-1) \times \frac{\mathcal{N}(z(t),\bar{z}_m(t\,|\,t-1),\bar{S}_m(t))}{\sum_{m'=1}^{M}\alpha_{m'}(t-1)\mathcal{N}(z(t),\bar{z}_{m'}(t\,|\,t-1),\bar{S}_{m'}(t))}. \quad (52)$$

The same weight update formula is also used in [12] and [18]. More sophisticated algorithms (correcting for the approximations made) can be found in [10].

In order to decentralize this algorithm, we must still decentralize the weight update. The cross terms of $\bar{S}_m(t)$ are of the form $\bar{H}_{i,m}\bar{P}_m(t\,|\,t-1)\bar{H}^T_{j,m}$, whereas the diagonal terms are given by $\bar{R}_{i,m} + \bar{H}_{i,m}\bar{P}_m(t\,|\,t-1)\bar{H}^T_{i,m}$. $\bar{R}_{i,m}$ takes into account the observation noise and the linearization noise. It is then usually big compared to variance due to the estimation spread $\bar{H}_{j,m}(t)\bar{P}_m(t\,|\,t-1)\bar{H}^T_{j,m}(t)$. From the Schwartz inequality, it is then also big compared to $\bar{H}_{j,m}(t)\bar{P}_m(t\,|\,t-1)\bar{H}^T_{k,m}(t)$. We can therefore make the approximation that the cross terms of $\bar{S}_m(t)$ are negligible, and we use the following approximation for the weight update:

$$\mathcal{N}\left(z(t);\bar{z}_m(t\,|\,t-1),\bar{S}_m(t)\right) \approx \prod_{s=1}^{S}\mathcal{N}(z_s(t);\bar{z}_{s,m}(t\,|\,t-1),\bar{S}_{s,m}(t)). \quad (53)$$

With this approximation, the complete algorithm can be summarized as follows.

1) At time t, all sensors share the same Gaussian mixture prior $p(x(t-1)\,|\,Z^{t-1})$ in (51).
2) Each sensor s performs the local computations for each Gaussian component m:
   a) Linearization of the local state-space using the mth kernel to get (36) and (48).
   b) Update step (44) and (45) to get the common predictions $\hat{y}_m(t\,|\,t-1)$ and $Y_m(t\,|\,t-1)$.
   c) Local information contributions computation: $i_{s,m}(t)$ from (49), $I_{s,m}(t)$ from (50), and the local weight contribution $\mathcal{N}(z_s(t);\bar{z}_{s,m}(t\,|\,t-1),\bar{S}_{s,m}(t))$.
3) The sensors share the information so that they all know the cumulative quantities corresponding to each Gaussian component m:
$$i_m(t) = \sum_{s=1}^{S} i_{s,m}(t), I_m(t) = \sum_{s=1}^{S} I_{s,m}(t), \text{and}$$
$$\prod_{s=1}^{S}\mathcal{N}(z_s(t);\bar{z}_{s,m}(t\,|\,t-1),\bar{S}_{s,m}(t)).$$
4) Each sensor computes the updated quantities for each Gaussian component m:
$$\begin{aligned} \hat{y}_m(t\,|\,t) &= \hat{y}_m(t\,|\,t-1) + i_m(t), Y_m(t\,|\,t) \\ &= Y_m(t\,|\,t-1) + I_m(t) \quad \text{and} \\ \alpha_m(t) &\propto \alpha_m(t-1)\prod_{s=1}^{S}\mathcal{N}(z_s(t);\bar{z}_{s,m}(t\,|\,t-1),\bar{S}_{s,m}(t)). \end{aligned}$$
5) Each sensor normalizes the weights of the mixture so that $\sum_{m=1}^{M}\alpha_m(t) = 1$.

If some weights of the mixture become zero or if the parameters of two mixtures become almost equal, then this algorithm would reduce to a simpler one, where the number of components is smaller. In this version of the algorithm, we have not taken this fact into account. Several possibilities are offered to gain some efficiency. The simpler one would be to drop the mixture components whose weights are smaller than some predefined threshold. Another possibility would be to recover a Gaussian mixture by drawing samples from the mixture and then to use a standard expectation-maximization (EM) algorithm [24] to compute the parameters of the new mixture. This technique is used in [12] and pushed a step further; the samples are considered to be in a particle filter and reweighted by using the measurement equation. The latter step is not really applicable to decentralized systems since the weighting process requires information from all sensors. One specificity of our decentralized algorithm is that we would like the nodes to share a common posterior density. In order to meet that requirement, the EM step should give the same results at each node. This could be done by applying the idea developed in [13]. If the sensor share a common seed (for example based on one of the prior weights), then the the samples are all equal, and so is the result of the EM step. Such a procedure could be applied every T steps or similarly to the sequential Monte Carlo resampling stage whenever a specific criterion is met. Future work will aim at defining a good measure of quality of the mixture.

III. Decentralized Collaborative Target Tracking

In the previous section, we introduced powerful yet computationally simple decentralized algorithms to make Bayesian inference in dynamic systems observed by several uncorrelated sensors. These algorithms can, however, not be straightforwardly used to solve the collaborative target tracking problem because of the so-called data-association problem. Indeed, in most cases, the tracking is performed in a cluttered environment where false detections are present and where the target might not be detected. The classical ways of dealing with the data association problem are the nearest-neighbor (NN) solution or

the probabilistic data association (PDA) [20]. None of those can readily be applied to the decentralized multisensor case.

The network is assumed to have been initialized so that it is possible for the sensor to communicate and share some cumulative statistics. We also assume that each sensor knows its position and that they share a common prior information (prior density, motion model).

A. Problem Formulation

The target is considered as a point-object moving in a two-dimensional plane. In this work, we consider a quite general, nonlinear motion model: the coordinated turn rate model. See [25] for an up-to-date survey of the available motion models. t is the discrete-time index, and $\Delta\tau$ denotes the length of a time step. This model assumes that the target moves with a nearly constant speed and a nearly constant unknown turn rate. We denote $x(t)$ as the state of the target and $v(t)$ as the corresponding motion noise. $x(t)$ represents the coordinates x_1, x_2, the velocities $\dot{x}_1, \dot{x}_2$, and the turn rate ρ: $x(t) \triangleq \{x_1(t), \dot{x}_1(t), x_2(t), \dot{x}_2(t), \rho(t)\}$. We have

$$x(t+1) = \begin{pmatrix} 1 & \frac{\sin\rho(t)\Delta\tau}{\rho(t)} & 0 & -\frac{1-\cos\rho(t)\Delta\tau}{\rho(t)} & 0 \\ 0 & \cos\rho(t)\Delta\tau & 0 & -\sin\rho(t)\Delta\tau & 0 \\ 0 & -\frac{1-\cos\rho(t)\Delta\tau}{\rho(t)} & 1 & \frac{\sin\rho(t)\Delta\tau}{\rho(t)} & 0 \\ 0 & \sin\rho(t)\Delta\tau & 0 & \cos\rho(t)\Delta\tau & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} x(t) + \begin{pmatrix} \frac{\Delta\tau^2}{2} & 0 & 0 \\ \Delta\tau & 0 & 0 \\ 0 & \frac{\Delta\tau^2}{2} & 0 \\ 0 & \Delta\tau & 0 \\ 0 & 0 & 1 \end{pmatrix} u(t) \quad (54)$$

where $u(t) \sim \mathcal{N}(0, \mathrm{diag}[\sigma_1^2, \sigma_2^2, \sigma_\rho^2])$.

At each time step, each sensor provides measurements that can be divided into two distinct sets: The first one consists of the measurement generated by the target if it is detected, and the second one is composed of the false detections generated by the clutter. The set of measurements acquired at time t by sensor s is denoted $Z_s(t) \triangleq \{z_s^i(t) \,|\, i = 1 \ldots n_s(t)\}$, where $n_s(t)$ is the number of measurements.

Many types of sensors provide measurements that are a function of the relative distance between the sensor and the sensed object (e.g., radar, acoustic sensors, sonar, etc.). Let $\{x_1^s, x_2^s\}$ be the position of sensor s and $d_s(x(t)) = [(x_2(t) - x_1^s)^2 + (x_2(t) - x_2^s)^2]^{(1/2)}$ be the distance between the sensor and the target. We consider a common example where sensors measure the power of a radio signal emitted by the object. The received power typically exponentially decays with the relative distance. In a logarithmic scale, the target-originated measurements are modeled by [26]

$$\begin{aligned} h_d(d) &= K - 10\eta \log_{10}(d) \\ z_s(t) &= h_d(d_s(x(t))) + w_s(t) \end{aligned} \quad (55)$$

where the measurement noise $w_s(t)$ accounts for the shadowing effects and other uncertainties; $w_s(t)$ is assumed to be a zero-mean independent and identically distributed (i.i.d.) Gaussian, i.e., $w_s(t) \sim \mathcal{N}(0, R_s)$, as in Section II, the sensor noises are assumed uncorrelated; K is the transmission power, and $\eta \in [2, 5]$ is the path loss exponent. These parameters depend on the radio environment, antenna characteristics, terrain, etc. Note that $\eta = 2$ corresponds to the free space transmission and serves as a lower limit. Furthermore, a sensor can typically provide measurements of a target only within a certain range. Therefore, a target could be detected only if $d_s(x_t) \in [d_{\min}, d_{\max}]$. In that case, we will denote by P_D the probability of detection, which is assumed known. Finally, it is assumed that the target can only provide a single measurement per sensor. In this model, the observation noise is additive and Gaussian, and the statistical linearization can thus be simplified, as mentioned previously.

The final probability of detection is thus given by $\bar{P}_D = P_D P[d_s(x_t) \in [d_{\min}, d_{\max}] \,|\, Z^{t-1}]$. There is no closed-form equation for $P[d_s(x_t) \in [d_{\min}, d_{\max}] \,|\, Z^{t-1}]$; however, we can use the sigma-points $\mathcal{X}_i(t \,|\, t-1)$ to approximate it by

$$\begin{aligned} &P[d_s(x_t) \in [d_{\min}, d_{\max}] \,|\, Z^{t-1}] \\ &\quad = E[\mathbf{1}_{d_s(x_t) \in [d_{\min}, d_{\max}]} \,|\, Z^{t-1}] \\ &\quad \approx \sum_i \mathbf{1}_{d_s(\mathcal{X}_i(t \,|\, t-1)) \in [d_{\min}, d_{\max}]}. \end{aligned}$$

The false detections are spurious measurements assumed to be uniformly distributed in the measurement area $\mathcal{A}_{\mathrm{meas}} = [h(d_{\max}), h(d_{\min})]$, whose volume is denoted as $V_{\mathrm{meas}} = |h(d_{\min}) - h(d_{\max})|$. The number of false detections $n_s^0(t)$ is typically generated by a Poisson distribution with parameter $\lambda V_{\mathrm{meas}}$, where λ is the number of clutter measurements per unit volume. Hence, we have

$$P\left(n_s^0(t) = k\right) = e^{-\lambda V_{\mathrm{meas}}} \frac{(\lambda V_{\mathrm{meas}})^k}{k!} \quad k = 0, 1, 2, \ldots \quad (56)$$

$$p_c(z) \sim \mathcal{U}_{\mathcal{A}_{\mathrm{meas}}}(z) \quad (57)$$

where $p_c(z)$ is the probability density function (p.d.f.) of any clutter measurement. If a gating procedure is used, the statistical properties of the clutter are the same, except that we only consider the area and volume of the gate $\mathcal{A}_{\mathrm{gate}}, V_{\mathrm{gate}}$ instead of $\mathcal{A}_{\mathrm{meas}}$ and V_{meas}.

B. Local Probabilistic Data Association

The NN algorithm constructs a validation gate centered around the predicted measurement and then uses, as the true one, the measurement closest to the predicted value with respect to the Mahalanobis distance weighted by $\bar{S}(t)$. As pointed out previously, we do not have access to the cross-terms of $\bar{S}(t)$, and the NN algorithm would thus either require a centralized processing or an approximation to dismiss the cross-terms of $\bar{S}(t)$. In this case, the aggregated NN is simply the aggregation of the local NN. It is, however, well known that the NN algorithm can perform very poorly in dense clutter noise and cause the filter to diverge. Extending the degree of approximation here would thus yield even worse results.

On the other hand, the PDA for multiple sensors [21] has a complexity growing exponentially in the number of sensors. In [27], the multisensor PDA is performed sequentially among the sensors at each time step. The advantages of the decentralized algorithm is then lost because we eventually go back to a sort of leader-based tracking scheme.

In this work, we propose to integrate a local probabilistic data association at each sensor into the decentralized information filters. All available measurements are then used to update the belief, but we ignore the correlation between the measurements. The computational need is thus kept low, and more importantly, the communication burden is similar to the approximate NN case. The inherent requirement of sensors networks can thus be met.

The data association indicator $a_s(t)$ designates which measurement (among the validated ones) arises from the target. $a_s(t) = 0$ means that the target has not been detected. The local estimate is given by

$$\begin{aligned}\hat{x}_s(t|t) &= E[x(t) \mid Z^{t-1}, Z_s(t)] \\ &= \sum_{i=0}^{n_s(t)} E\left[x(t) \left| Z^{t-1}, z_s^i(t)\right.\right] p\left(a_s(t) = i | Z^{t-1}, Z_s(t)\right).\end{aligned} \tag{58}$$

For $a_s(t) = i \neq 0, E[x(t) \mid Z^{t-1}, z_s^i(t)]$ is given by (13), and for $a_s(t) = 0$, $E[x(t) \mid Z^{t-1}, z_s^0(t)]$ is simply the prediction from (11); z_s^0 simply represents no information. We therefore have

$$\hat{x}_s(t \mid t) = \hat{x}(t \mid t-1) + \bar{W}_s(t) \sum_{i=1}^{n_s(t)} \beta_s^i(t) \nu_s^i(t) \tag{59}$$

where $\beta_s^i(t) = p(a_s(t) = i \mid Z^{t-1}, Z_s(t))$ can be found in [20]. Similarly, we get

$$P_s(t \mid t) = P(t \mid t-1) - \bar{W}_s(t)[\bar{S}_s(t) - B_s(t)]\bar{W}_s^T(t) \tag{60}$$

where $B_s(t) = \beta_s^0(t)\bar{S}_s(t) + \sum_{i=1}^{n_s(t)}$, and $\nu_s(t) = \sum_{i=1}^{n_s(t)} \beta_s^i(t)\nu_s^i(t)$ is the weighted innovation.

In order to find the information form of the local probabilistic data association above, we would like to find $\tilde{R}_s(t)$ and $\zeta_s(t)$ such that (59) and (60) have the the same form as the usual Kalman filter

$$P_s(t \mid t) = P(t \mid t-1) - \tilde{W}_s(t)\tilde{S}_s(t)\tilde{W}_s^T(t) \tag{61}$$

$$\hat{x}_s(t \mid t) = \hat{x}(t \mid t-1) + \tilde{W}_s(t)[\zeta_s(t) - \bar{H}_s(t)\hat{x}(t \mid t-1)] \tag{62}$$

where

$$\begin{aligned}\tilde{W}_s(t) &= P(t \mid t-1)\bar{H}_s^T\tilde{S}_s^{-1}(t) \\ \tilde{S}_s(t) &= \bar{H}_s(t)P(t \mid t-1)\bar{H}_s^T(t) + \tilde{R}_s(t).\end{aligned}$$

The classical derivation of the information filter will then lead to an easily decentralized algorithm. This is achieved (cf. the Appendix) by choosing

$$\tilde{R}_s(t) = \bar{R}_s(t) + \left[B_s^{-1}(t) - \bar{S}_s^{-1}(t)\right]^{-1} \tag{63}$$

$$\begin{aligned}\zeta_s(t) = &\left[I + \left(B_s^{-1}(t) - \bar{S}_s^{-1}(t)\right)^{-1} \bar{S}_s^{-1}(t)\right] \nu_s(t) \\ &+ \bar{H}_s(t)x(t \mid t-1)\end{aligned} \tag{64}$$

which implies $\tilde{S}_s(t) = \bar{S}_s(t) + [B_s^{-1}(t) - \bar{S}_s^{-1}(t)]^{-1}$ and $\tilde{W}_s(t) = P(t \mid t-1)\bar{H}_s^T(t)\tilde{S}_s^{-1}(t)$. Going to the information form, we get

$$i_s(t) = \bar{H}_s^T(t)\tilde{R}_s^{-1}(t)\zeta_s(t) \tag{65}$$

$$I_s(t) = \bar{H}_s^T(t)\tilde{R}_s^{-1}(t)\bar{H}_s(t). \tag{66}$$

Based on our simulation results, we make the approximation that (49) and (50) and (44) and (45) can be used, and therefore, the extension of the decentralized algorithm becomes trivial. Because that update technique incorporates the loss of information due to the association uncertainty and uses all available measurements, it performs better than an NN association while keeping the same complexity.

The same procedure is used when dealing with the Gaussian mixture. The only additional change resides in the approximate weight update (52), which becomes

$$\begin{aligned}\alpha_m(t) \propto \alpha_m(t-1)p(Z_s(t) \mid Z^{t-1}) \propto \alpha_m(t-1) \sum_{a_s(t)} \\ p(Z_s(t) \mid Z^{t-1}a_s(t))p(a_s(t), n_s(t) \mid Z^{t-1})\end{aligned} \tag{67}$$

where $p(Z_s(t) \mid Z^{t-1}, a_s(t))$ and $p(a_s(t), n_s(t) \mid Z^{t-1})$ can be found in [20].

IV. Simulation Results

To illustrate the performance of the proposed sigma-point information filters, simulations are performed under several different conditions. First, the decentralized sigma-point information filters are compared with the decentralized extended information filter (DEIF) from [4] without any data association uncertainty. Then, our local PDA algorithm is compared with a local NN approach (both using a sigma-point information filter) under light and heavy clutter noise.

The simulated track is kept identical for each simulation. We can see in Fig. 1, for example, that the nonlinearity of the state-transition model is clearly shown. There is a first period where the target moves almost on a straight line (with sign-changing curvature) and then it begins to turn. The motion and sensing model are taken from Section III. The sensor network consists of 15 sensors randomly scattered over a field of 80 m $\times$ 60 m. We assume that there is no communication loss and that the sensors are synchronized. For all simulations, we take the following parameters. The transmission power for each sensor is $K = 9$ dBm, and the path loss index is $\eta = 3$. The time step is chosen to be $\Delta\tau = 1$ sec. The process noise covariance is given by $Q = 0.001^2 I_3$, and the observation noise variance (shadowing effects and other uncertainties) is $R = 5$. The algorithms are compared with the same sets of measurements. The criterion for comparison is the mean-squared error (MSE) on the position of the target. The 3-sigma ellipsoids are shown every 30 time steps.

The prior information is as follows. The prior estimate is $\hat{x}(0 \mid 0) = x(0) + x_{\text{bias}}$, where x_{bias} is a strong

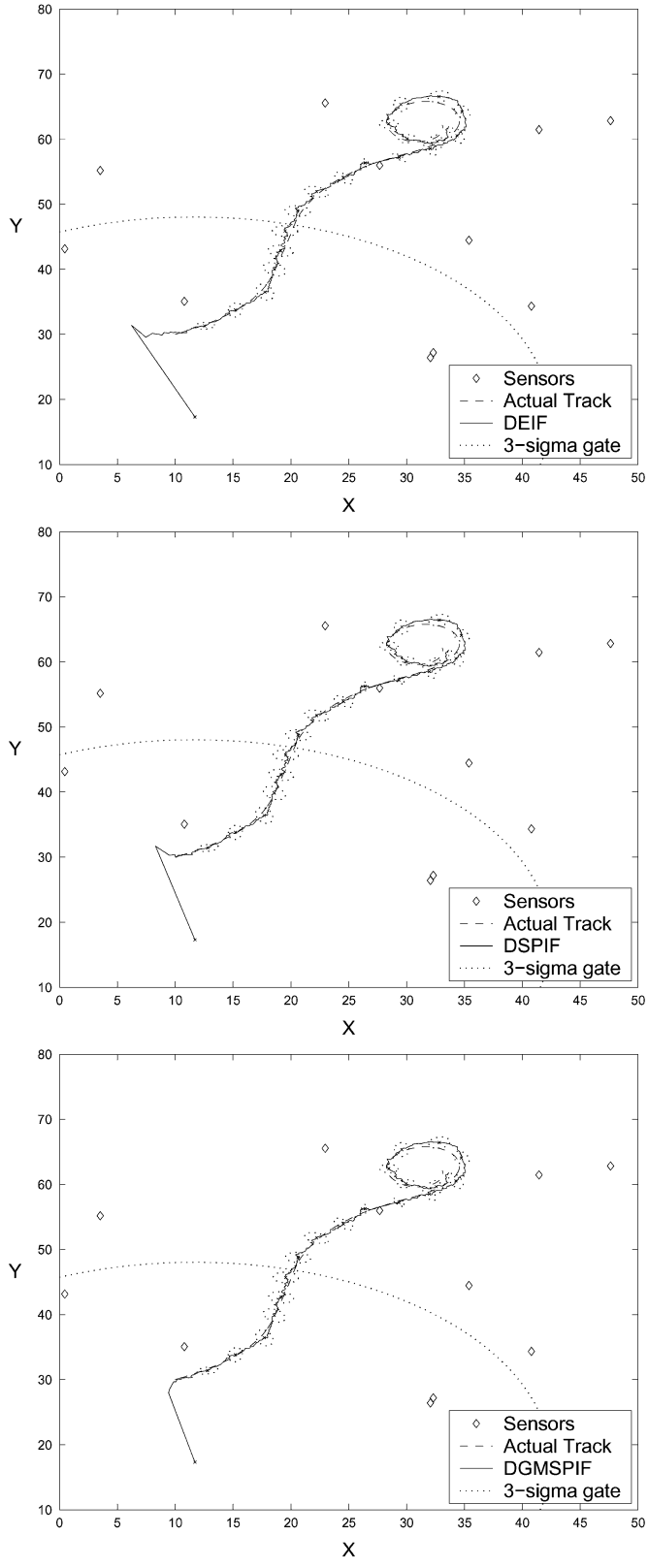


Fig. 1. Actual and estimated trajectories with $d_{\min} = 0$ and $d_{\max} = \infty$. One sample run.

bias on the position only, drawn according to a uniform distribution on a square of length 30 m centered on $[0, 0]$. The Gaussian mixture is composed of five equally weighted Gaussian distributions with the same covariances $P_m(0\,|\,0) = \text{diag}(5^2, 5^2, 0.005^2, 0.005^2, 0.000\,05^2)$ and means $\hat{x}(0\,|\,0), \hat{x}(0\,|\,0) \pm [10, 10, 0, 0, 0]^T$ and $\hat{x}(0\,|\,0) \pm [-10, 10, 0, 0, 0]^T$. The parameters of the Gaussian density for the other algorithms are chosen to provide the same mean and same global covariance.

A. Simulations Without Data Association Ambiguity

We compare the proposed algorithms of Section II with the decentralized extended information filter. The first set of simulations illustrate the performance of the different algorithm in a usual nonlinear state-space where the observation function is smooth. All observations are given by (55).

Fig. 1 shows the result of the tracking algorithms for a single run. The three possible estimates are quite accurate, as expected. The performance of the three algorithms is shown in Fig. 3(a). We can see that the sigma-point information filters perform similarly. In this case, the use of the DSPIF is thus advocated. Both proposed filters perform better that the DEIF. All filters seem to have bounded errors.

The second scenario considered is still a comparison with the DEIF. The observation function is chosen to be less smooth than previously in the sense that it mimics the fact that the target can only be detected if $d_s(x_t) \in [d_{\min}, d_{\max}]$. Therefore, we choose h_s such that

$$h_s(x_t) = \begin{cases} h_d(d_{\min}), & \text{if } d_s(x_t) < d_{\min} \\ h_d(d_s(x_t)), & \text{if } d_s(x_t) \in [d_{\min}, d_{\max}] \\ h_d(d_{\max}), & \text{if } d_s(x_t) > d_{\max}. \end{cases}$$

The values of those parameters are taken as $d_{\min} = 1$ m and $d_{\max} = 50$ m. The measurement range becomes approximatively $[-41$ dBm, 9 dBm$]$.

Fig. 2 shows the result of the tracking algorithms for a single run. The three estimates for this run show good tracking results. However, when the performance of the three algorithms is compared on 50 runs, we can see in Fig. 3(b) that the DEIF has very large errors compared with the other two algorithms; this can be explained by the fact that the DEIF sometimes diverges, whereas the sigma-points filters remain accurate. Both sigma-point information filters perform well, but the DGMSPIF shows smaller MSE than the DSPIF during the initiation of the filter.

B. Simulations Including Data Association Ambiguity

The simulation results discussed in this section illustrate the performance of the proposed algorithms in more realistic tracking scenarios. The data association ambiguity is now considered. The previous simulations showed a divergence tendency for the DEIF, which is even more present in the case now considered. Hence, the results shown here do not include the DEIF. We are now concerned about the performance of the proposed local probabilistic data association (LPDA) of Section III. Therefore, we make a comparison of the DSPIF using a nearest-neighbor rule (NN-DSPIF) and the proposed algorithm using the LPDA (LPDA-DSPIF and LPDA-DGMSPIF) under both a lightly and heavily cluttered environment.

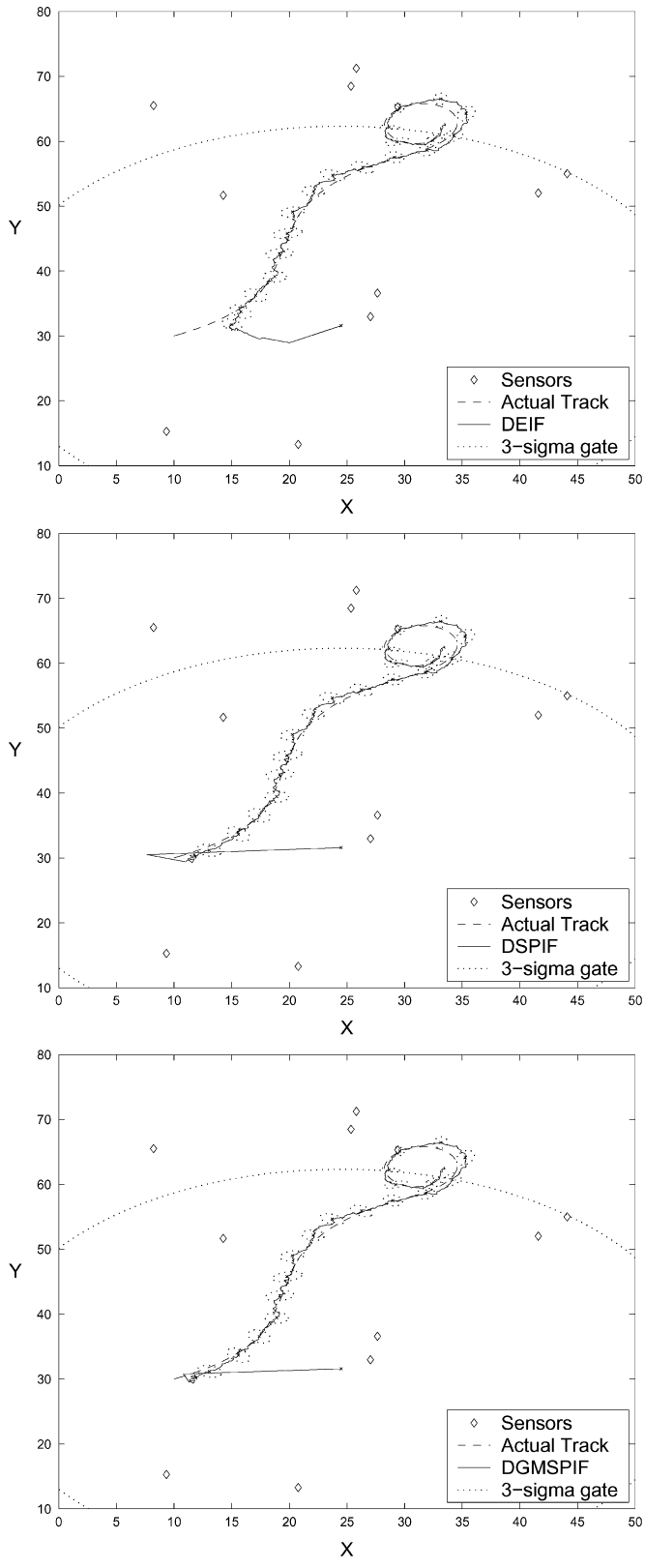


Fig. 2. Actual and estimated trajectories with $d_{\min} = 1$ and $d_{\max} = 50$. One sample run.

The probability of detection is $P_D = 0.9$ when the target is in the range [1 m, 50 m] of the sensor and 0 otherwise. The classical gating procedure is also used. Only the measurements lying close enough to the predicted measurement

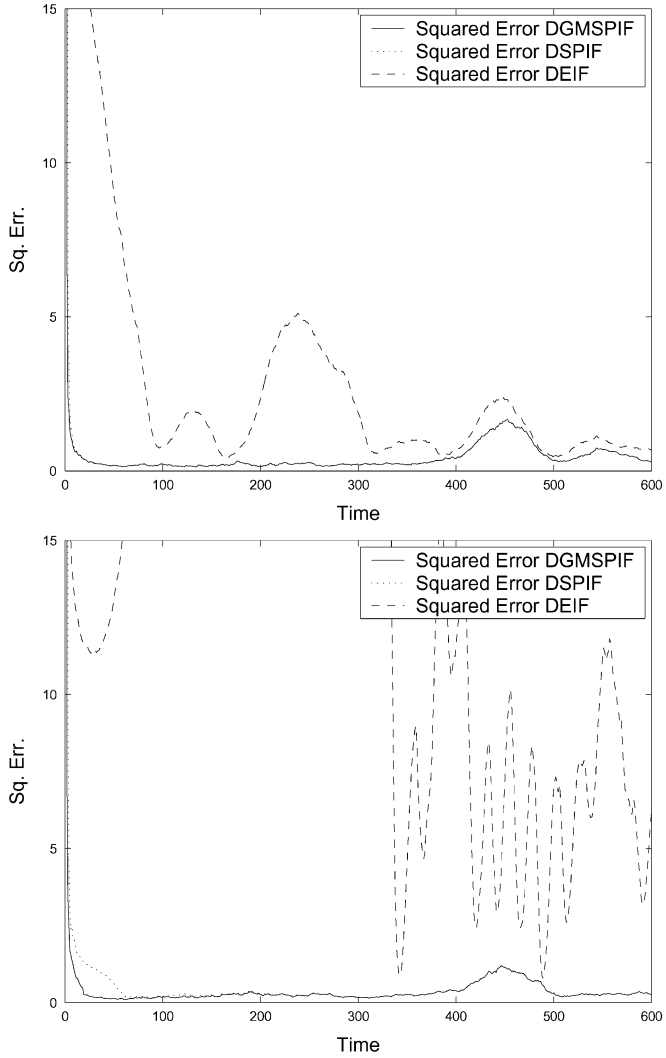


Fig. 3. Mean-square error on the position for 50 runs without data association ambiguity. (a) $d_{\min} = 0, d_{\max} = \infty$. (b) $d_{\min} = 1, d_{\max} = 50$.

are considered. Typically, the gate is a ℓ-sigma ellipsoid $\mathcal{A}_{\text{gate}} = \{z_s^i(t) \,|\, \nu_s^{iT} \bar{S}(t) \nu_s^i(t) \leq \ell\}$ based on the innovation of the measurement vector $\nu_s^i(t) = z_s^i(t) - \hat{z}_s(t \,|\, t-1)$. ℓ is chosen so that there is a probability P_G (here, we make the typical choice $P_G = 0.99$) that the true measurement lies in the validation gate.

The first scenario shown in Fig. 4 is the light clutter case. The parameter of the Poisson distribution for the entire measurement volume is $\lambda V_{\text{meas}} = 0.5$, and because $\bar{S} > R$, by considering only the observation noise, we also have $\lambda V_{\text{gate}} > 0.13$ when the gate $\mathcal{A}_{\text{gate}}$ is included in $\mathcal{A}_{\text{meas}}$. The NN as well as the LPDA-based algorithm show good results in Fig. 4, and this is confirmed by the MSE plot in Fig. 6(a). As expected in this case, the difference between the algorithms is minor, even if we can see in the most difficult tracking part (after $t = 500$ s) that the LPDA-DGMSPIF performs better than the LPDA-DSPIF itself, which performs better than the NN-DSPIF.

The real advantage of the LPDA is shown in the dense clutter case. We now choose $\lambda V_{\text{meas}} = 5$, which implies $\lambda V_{\text{gate}} > 1.3$ by taking only the observation noise into account. In the single

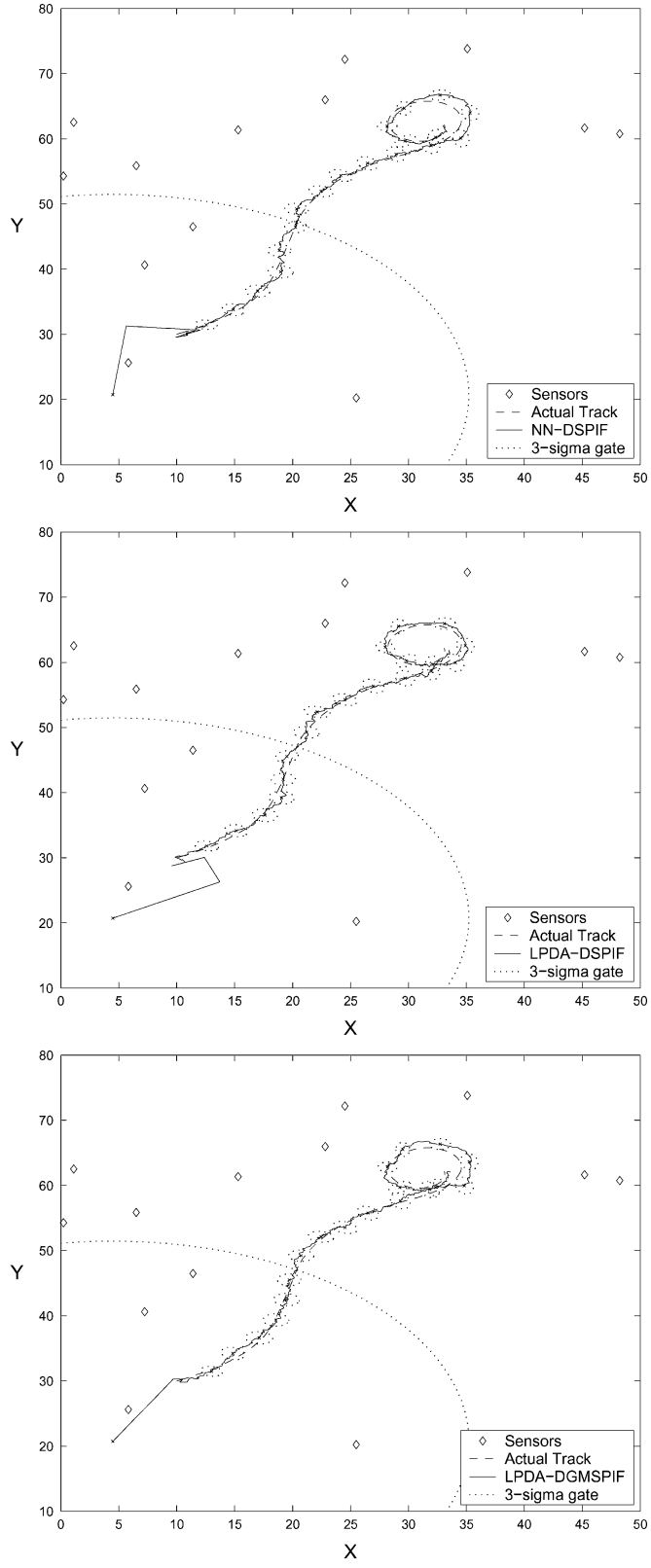


Fig. 4. Actual and estimated trajectories with $\lambda V_{\text{meas}} = 0.5$. One sample run.

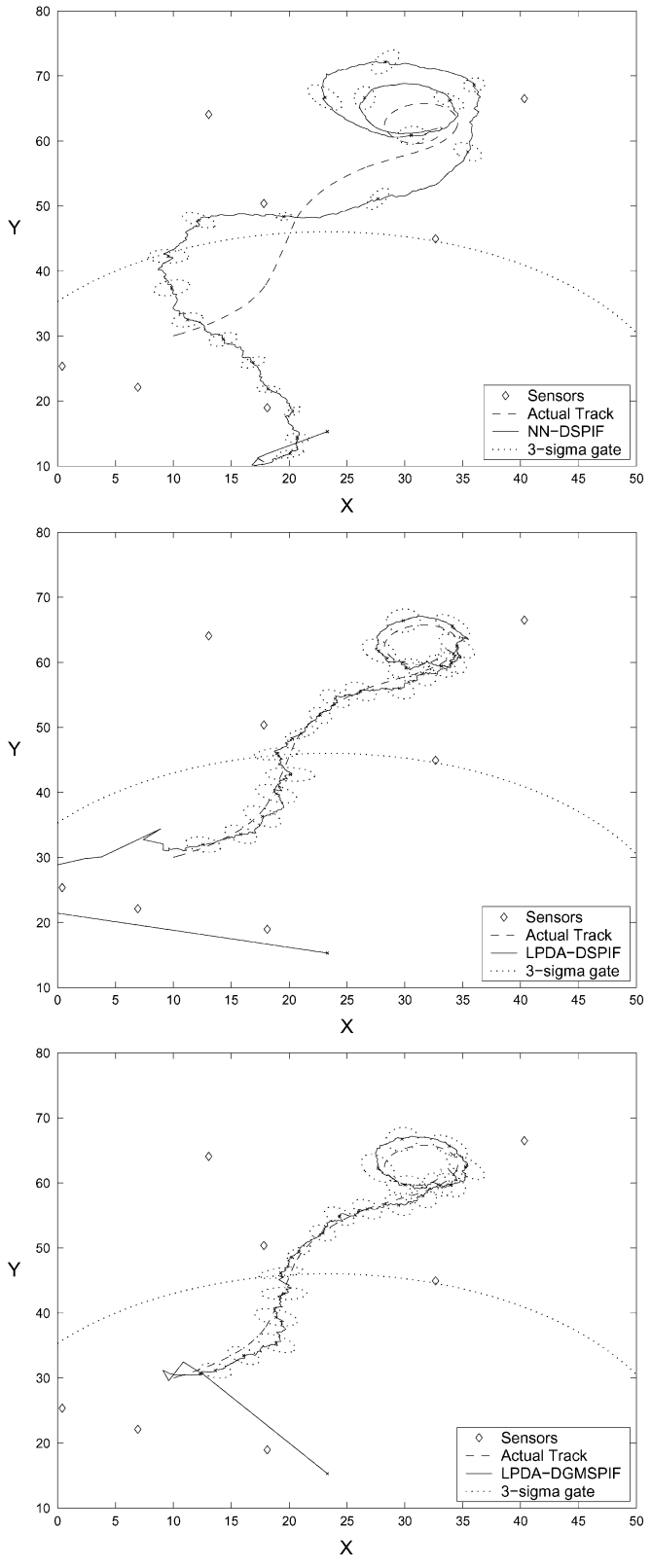


Fig. 5. Actual and estimated trajectories with $\lambda V_{\text{meas}} = 5$. One sample run.

run estimates shown in Fig. 5, we can see that the estimate of the NN-DSPIF does not stand up to the comparison with the other two estimates, even if the error seems bounded. On the other hand, with the same set of measurements, the LPDA-based algorithm provides accurate results. The MSE plot in Fig. 6(b) confirms this analysis and shows that the NN algorithm diverges, whereas both LPDA-based algorithms provide bounded errors. Furthermore, in this difficult case, the LPDA-DGMSPIF shows its superiority to the LPDA-DSPIF.

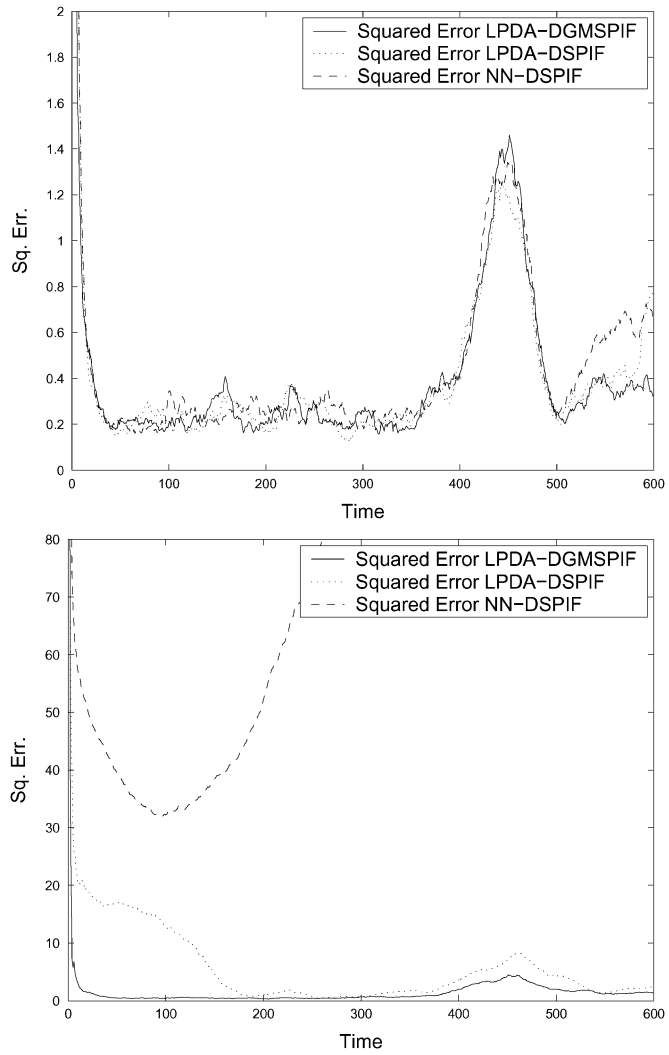


Fig. 6. Mean-square error on the position for 50 runs with (a) $\lambda V_{\text{meas}} = 0.5$. (b) $\lambda V_{\text{meas}} = 5$.

V. Conclusion

We have presented a class of strictly decentralized algorithms for Bayesian filtering in nonlinear dynamic state spaces. Our approach makes use of the information filter framework and the statistical linearization methodology to provide accurate and robust collaborative inference algorithms under Gaussian approximation. The extension to multimodal density tracking is made by using a Gaussian-mixture model and a bank of parallel sigma-point information filters. These schemes are shown to be effective for in-network signal processing since the complexity is that of a local Kalman filter running at each activated node. The number of communications needed can be made linear with the number of activated sensors. The proposed sigma-point information filters have then been adapted to the target tracking problem. A local probabilistic data association rule was developed to fit the information space inference and keep the complexity equivalent to that of a filter using an NN rule. The performance of this algorithm was assessed by simulations.

One topic of future research is to tackle the problem of sensor selection. Indeed, the proposed scheme can take advantage of the leader-based processing techniques where, at each time step, only a subset of nodes is active. The simplest approach would be to choose in the active set the expected less informative node, have it activate the expected most informative node from the idle nodes, have it hand off its belief to this node, and then let it go to an idle state. An efficient way of developing this heuristic would certainly provide advantageous results.

Appendix
Information Form of the Local PDA

In this Appendix, we will derive (63). We first notice that

$$\begin{aligned}\tilde{W}_s(t) &= P(t\,|\,t-1)\bar{H}_s^T(t)\bar{S}_s^{-1}(t)\bar{S}_s(t)\tilde{S}_s^{-1}(t)\\ &= \bar{W}_s(t)\bar{S}_s(t)\tilde{S}_s^{-1}(t).\end{aligned}$$

Therefore

$$\begin{aligned}&\tilde{W}_s(t)\tilde{S}_s(t)\tilde{W}_s^T(t)\\ &\quad= \bar{W}_s(t)\bar{S}_s(t)\tilde{S}_s^{-1}(t)\tilde{S}_s(t)\tilde{S}_s^{-1}(t)\bar{S}_s(t)\bar{W}_s^T(t)\\ &\quad= \bar{W}_s(t)\bar{S}_s(t)\tilde{S}_s^{-1}(t)\bar{S}_s(t)\bar{W}_s^T(t)\end{aligned}$$

and (61) is satisfied if $\bar{S}_s(t)\tilde{S}_s^{-1}(t)\bar{S}_s(t) = \bar{S}_s(t) - B_s(t)$. This can be reexpressed as $\tilde{S}_s^{-1}(t) = \bar{S}_s^{-1}(t) - \bar{S}_s^1(t)B_s(t)\bar{S}_s^{-1}(t)$. Using the matrix inversion lemma, we get

$$\begin{aligned}\tilde{S}_s(t) &= \bar{S}_s(t) - \bar{S}_s(t)\bar{S}_s^{-1}(t)\\ &\quad\times\left[\bar{S}_s^{-1}(t)\bar{S}_s(t)\bar{S}_s^{-1}(t) - B_s^{-1}(t)\right]^{-1}\bar{S}_s^{-1}(t)\bar{S}_s(t)\\ &= \bar{S}_s(t) + \left[B_s^{-1}(t) - \bar{S}_s^{-1}(t)\right]^{-1}. \qquad (68)\end{aligned}$$

In order to satisfy (62), we need

$$\begin{aligned}0 &= \tilde{W}_s(t)[\zeta_s(t) - \bar{H}_s(t)\hat{x}(t\,|\,t-1)] - \bar{W}_s(t)\nu_s(t)\\ &= \bar{W}_s(t)\left[\bar{S}_s(t)\tilde{S}_s^{-1}(t)(\zeta_s(t) - \bar{H}_s(t)\hat{x}(t\,|\,t-1)) - \nu_s(t)\right].\end{aligned}$$

This requirement is met by choosing $\zeta_s(t)$ such that

$$\begin{aligned}\zeta_s(t) &= \bar{H}_s(t)\hat{x}(t\,|\,t-1) + \tilde{S}_s(t)\bar{S}_s^{-1}(t)\nu_s(t)\\ &= \left[I + \left(B_s^{-1}(t) - \bar{S}_s^{-1}(t)\right)^{-1}\bar{S}_s^{-1}(t)\right]\\ &\quad\times\nu_s(t) + \bar{H}_s(t)\hat{x}(t\,|\,t-1). \qquad (69)\end{aligned}$$

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] J. Liu, J. Reich, and F. Zhao, "Collaborative in-network processing for target tracking," in *EURASIP J. Appl. Signal Processing*, Mar. 2003, pp. 378–391.

[3] M. Chu, H. Haussecker, and F. Zhao, "Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor network," *Int. J. High Perf. Comput. Appl.*, vol. 16, no. 3, pp. 293–314, Mar. 2002.

[4] A. G. O. Mutambara, *Decentralized Estimation and Control for Multisensor Systems*. Boca Raton, FL: CRC, 1998.

[5] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*: Prentice-Hall, 1979.

[6] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1996.

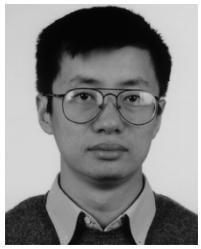
[7] J. Manyika and H. Durrant-Whyte, *Data Fusion and Sensor Management: A Decentralized Information-Theoretic Approach*. Englewood Cliffs, NJ: Prentice-Hall, 1994.

[8] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformations of means and covariances in filters and estimators," *IEEE Trans. Autom. Control*, vol. 45, no. 3, pp. 477–482, Mar. 2000.

[9] S. J. Julier and J. K. Uhlmann, "The scaled unscented transformation," in *Proc. IEEE Amer. Control Conf.*, May 2002, pp. 4555–4559.
[10] K. Ito and K. Xiong, "Gaussian filters for nonlinear filtering problems," *IEEE Trans. Autom. Control*, vol. 45, no. 5, pp. 910–927, May 2000.
[11] T. Lefebvre, H. Bruyninckx, and J. De Schutter, "Comment on "a new method for the nonlinear transformation of means and covariances in filters and estimators"," *IEEE Trans. Autom. Control*, vol. 47, no. 8, pp. 1406–1408, Aug. 2002.
[12] R. van der Merwe and E. Wan, "Sigma-point Kalman filters for probabilistic inference in dynamic state-space models," in *Proc. Workshop Adv. in Machine Learning*, Montreal, QC, Canada, Jun. 2003.
[13] M. J. Coates, Distributed particle filtering for sensor networks, in Proc. Int. Symp. Inf. Process. Sensor Networks, Berkeley, CA, Apr. 2004.
[14] M. Rosencrantz, G. Gordon, and S. Thrun, "Decentralized sensor fusion with distributed particle filters," in *Proc. Conf. Uncertainty Artificial Intell.*, Acapulco, Mexico, 2003.
[15] A. Doucet, S. J. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for bayesian filtering," *Statist. Comput.*, vol. 10, no. 3, pp. 197–208, 2000.
[16] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2001.
[17] H. W. Sorenson and D. L. Alspach, "Recursive Bayesian estimation using Gaussian sums," *Automatica*, vol. 7, no. 4, pp. 465–479, July 1971.
[18] D. L. Alspach and H. W. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations," *IEEE Trans. Autom. Control*, vol. AC-17, no. 4, pp. 439–448, Aug. 1972.
[19] J. Kotecha and P. M. Djuric, "Gaussian sum particle filtering," *IEEE Trans. Signal Process.*, vol. 51, no. 10, pp. 2602–2612, Oct. 2003.
[20] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. New York: Academic, 1988.
[21] K. C. Chang, C. Y. Chong, and Y. Bar-Shalom, "Joint probabilistic data association in distributed sensor networks," *IEEE Trans. Autom. Control*, vol. AC-31, no. 10, pp. 889–897, Oct. 1986.
[22] H. V. Poor, *An Introduction to Signal Detection and Estimation*, Second ed. New York: Springer-Verlag, 1994.
[23] C. Y. Chong, "Hierarchical estimation," in *Proc. 2nd MIT/ONR Workshop C3*, Jun. 1979.
[24] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Statist. Soc. (B)*, vol. 39, no. 1, pp. 1–38, 1977.
[25] X. R. Li and V. P. Jilkov, "A survey of maneuvering target tracking: Dynamic models," in *Proc. SPIE Signal Data Process. Small Targets*, vol. 4048-22, Apr. 2000.
[26] T. Rappaport, *Wireless Communications: Principles and Practice*, Second ed. Englewood Cliffs, NJ: Prentice-Hall, 2002.
[27] A. Houles and Y. Bar-Shalom, "Multisensor tracking of a maneuvering target in clutter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 25, no. 2, pp. 176–189, Mar. 1989.

Tom Vercauteren (S'05) graduated from Ecole Polytechnique, Paris, France, in 2003 and received the M.S. degree from the Department of Electrical Engineering, Columbia University, New York, NY, in 2004. He is now working toward the Ph.D. degree with INRIA, Sophia-Antipolis, France.

His research interests are in the area of statistical signal processing.



Xiaodong Wang (S'98–M'98–SM'04) received the B.S. degree in electrical engineering and applied mathematics (with the highest honor) from Shanghai Jiao Tong University, Shanghai, China, in 1992, the M.S. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, in 1995, and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, in 1998.

From July 1998 to December 2001, he was an Assistant Professor with the Department of Electrical Engineering, Texas A&M University, College Station. From January 2002, he has been on the faculty of the Department of Electrical Engineering, Columbia University, New York, NY. His research interests fall in the general areas of computing, signal processing, and communications. He has worked in the areas of digital communications, digital signal processing, parallel and distributed computing, nanoelectronics and quantum computing, and has published extensively in these areas. Among his publications is a recent book entitled *Wireless Communication Systems: Advanced Techniques for Signal Reception* (Prentice-Hall, Upper Saddle River, NJ: 2003). His current research interests include wireless communications, Monte Carlo-based statistical signal processing, and genomic signal processing.

Dr. Wang received the 1999 NSF CAREER Award and the 2001 IEEE Communications Society and Information Theory Society Joint Paper Award. He currently serves as an Associate Editor for the IEEE TRANSCTIONS ON COMMUNICATIONS, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON SIGNAL PPROCESSING, and the IEEE TRANSACTIONS ON INFORMATION THEORY.