Joint Multiple Target Tracking and Classification in Collaborative Sensor Networks

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

***Abstract*—We address the problem of jointly tracking and classifying several targets within a sensor network where false detections are present. In order to meet the requirements inherent to sensor networks such as distributed processing and low-power consumption, a collaborative signal processing algorithm is presented. At any time, for a given tracked target, only one sensor is active. This leader node is focused on a single target but takes into account the possible existence of other targets. It is assumed that the motion model of a given target belongs to one of several classes. This class-target dynamic association is the basis of our classification criterion. We propose an algorithm based on the sequential Monte Carlo (SMC) filtering of jump Markov systems to track the dynamic of the system and make the corresponding estimates. A novel class-based resampling scheme is developed in order to get a robust classification of the targets. Furthermore, an optimal sensor selection scheme based on the maximization of the expected mutual information is integrated naturally within the SMC target tracking framework. Simulation results are presented to illustrate the excellent performance of the proposed multitarget tracking and classification scheme in a collaborative sensor network.**

***Index Terms*—Collaborative signal processing, multitarget tracking, sensor networks, sequential Monte Carlo (SMC).**

I. Introduction

THE CONVERGENCE of recent developments in microelectromechanical systems (MEMS), microprocessors and ad hoc networking protocols have enabled low-power and low-cost sensor nodes endowed with sensing and processing capabilities to collaborate and achieve large tasks [1]. Typical applications of such sensor networks are event detection, event identification, and location sensing [2]. This paper focuses on the problem of jointly tracking and classifying several targets evolving within densely scattered sensor nodes. On the one hand, multiple target tracking tackles the issue of sequentially estimating the state of a possibly varying number of objects; and on the other hand, classification deals with the identification of those objects down to a given class. Because the number of targets can vary, we are handling three closely coupled subjects: target detection, tracking, and classification. Considering the strong interrelations existing between those, it is natural to address them jointly. Indeed, the class of a target defines its motion characteristics which are essential for accurate tracking. Besides, the observed target dynamic can be used to distinguish the type of the tracked object, and naturally, a change in the number of targets implies a modification of the tracking and classification procedures. In this paper, classification is based solely on the motion model of the targets (as opposed to the case in which two targets may differ in other characteristics, say audio spectrum, rather than their dynamical characteristics).

Typical algorithms dealing with multitarget tracking are very computationally complex and generally require a centralized computation based on the measurements available from all sensors [3], [4]. An important characteristic of sensor networks is their ability to cooperate among densely and randomly deployed sensor nodes [5]. Another significant feature is the low-power consumption requirement [1]. Sensor nodes carry limited, generally irreplaceable, power sources. It is, thus, of great importance to develop localized algorithms, where only a subset of the nodes are activated and are responsible for data fusion, instead of sending their raw measurements to an information fusion node. These requirements led us to use a leader-based tracking scheme [2], [5], where for each tracked target only one sensor, the leader-node, is active.

Sensor nodes can typically only sense objects in their close neighborhood. We, thus, make the reasonable assumption that no more than two targets can be at the same time in the field covered by a sensor. From the point of view of a leader, the problem becomes that of tracking and classifying a single target with a possible interfering target. The complex problem of multitarget tracking can, thus, be tackled through sensor collaboration by dividing it into several easier, localized tasks.

In this paper, the problem of information fusion and sensor selection is solved within the Bayesian framework. No closed-form solution for the posterior distribution of the target states is available and, therefore, sequential Monte Carlo (SMC) methods are employed to approximate the filtering density. In order to model the varying number of targets, we will make use of a jump Markov system (JMS) [6], [7].

The focus of this work is on the processing of the measurements to perform the joint tracking and classification as well as sensor management. The communication aspects are out of the scope of this paper. We, therefore, assume that the network has been initialized so that each node has the knowledge of its own position, its neighbors' identities and their positions.

The first contribution of this paper is to extend the (single target) leader-based tracking scheme proposed in [2] and [5] to the multiple target scenario by introducing a lower complexity version of the SMC filtering of JMS proposed in [6]. The second contribution is a class-based resampling procedure to avoid the loss of plausible classification hypotheses during the early stage

Manuscript received December 1, 2003; revised July 30, 2004. This work was supported in part by the U.S. National Science Foundation (NSF) under Grant CCR-0225826 and Grant DMS 0244583, and in part by the U.S. Office of Naval Research (ONR) under Grant N00014-03-1-0039. This paper was presented in part at the 2004 IEEE Symposium on Information Theory, Chicago, IL, June 27–July 2.

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

Digital Object Identifier 10.1109/JSAC.2005.843540

of the tracking. The third contribution is a SMC implementation of an optimal sensor selection scheme.

The remainder of this paper is organized as follows. In Section II, we provide a framework for multiple target tracking and classification in a collaborative sensor network. The general principle of sequential Monte Carlo inference is briefly reviewed in Section III. The issue of classification is discussed in Section IV, where we present the joint tracking and classification SMC algorithm for a single target. The multitarget tracking and classification algorithm is developed in Section V. In Section VI, we discuss the sensor selection scheme. Simulation results are presented in Section VII. Section VIII concludes the paper.

II. System Description and Problem Formulation

We assume that in a sensor network, the randomly deployed sensor nodes are able to collect data, process it, and route information back to the *sink* that dispatches the data to the end user. One of the main concern of any signal processing algorithm for sensor networks is to make efficient use of the limited available power resources. Each node should remain idle unless queried to perform a specific task [1].

A. Leader-Based Tracking in Sensor Networks

We consider a leader-based tracking scheme—also denoted as information-driven sensor querying (IDSQ) [2], [5]. For each tracked target and at each time step, only one sensor is active. The information is passed from node to node so that a single sensor, the leader node, is responsible for the tracking of a given target. Even if a leader is focused on a single target, unless the other targets are far away, it cannot consider them as noise because the statistical properties of the measurements arising from those other targets are identical to those of the tracked target (as opposed to spurious measurements) [3]. Therefore, a leader node needs to take into account other targets as soon as they appear in its covered field. As mentioned previously, it is assumed that in the limited range covered by a sensor, no more than two targets can be present. We further assume that the primary target is always within the range of its leader node. Each leader will, thus, have to deal with the simplest nontrivial multiple target tracking and classification problem. From now on, we can focus on the point of view of a single leader node.

An interesting problem in this setting is that of track initiation and track maintenance which is not covered in this work. We refer to [8] for a possible solution to these problems. In this paper, we assume that priori information is given to initiate the leader-based tracking scheme.

Each current leader node performs the following operations at each time instant.

- Step 1) The measurements are retrieved.
- Step 2) The posterior density (state and class of the two targets and probability of being in the field for the interfering target) is updated (cf. Sections IV and V).
- Step 3) A next leader node (for the the primary target) is chosen (cf. Section VI).
- Step 4) The belief state is handed over to this chosen node, which becomes the current leader node.
- Step 5) The node gets back to an idle state.

Such a leader-based scheme has several advantages that are particularly attractive for collaborative sensor network applications: only local computations are involved, no global knowledge is assumed, there is no need for centralized control, and it is perfectly scalable.

B. Multiple Target Tracking and Classification

When performing joint tracking and classification, we aim at giving a good estimate of the state of a target (position, velocity, and class). We consider a model-based target tracking method [9], where the target motions and the observations can be represented by state-space models. The state of the system is the concatenation of the states of the targets.

We consider class-dependent motion models where each target is considered as a point-object moving according to its dynamic in a two-dimensional plane. Those motion models are essential to any model-based tracking algorithm and, thus, need to be well fit to the tracked targets.

Let r_t denotes the number of targets at time t within the field of the considered leader node. Let T_t be the set of active targets at time t. From our assumptions, $T_t = \{1\}$ if $r_t = 1$ $T_t = \{1, 2\}$ if $r_t = 2$. The first target is always the tracked target and the second a generic interfering one. If $r_t = r_{t-1} = 2$, it is assumed that the second target remains the same (i.e., the case where the second target disappears, while a new target appears during the same time step is not considered). We denote by $X_t = \{x_{t,i}, \gamma_i\}_{i \in T_t}$ the state of the targets at time t, where $\{x_{t,i}, \gamma_i\}$ stands for the state (position, velocity, and class) of the i^{th} target. The system at time t is, thus, characterized by the vector (X_t, r_t), where r_t is introduced to emphasize the dependence of X_t on the number of targets. Conditioned on the number of targets at time t and $t+1$, the system dynamic model is described by $p(X_t|X_{t-1}, r_t, r_{t-1})$.

We make the common assumption that each target moves independently from the other according to a Markovian transition dynamic. This dynamic depends on the class γ_i of the ith target. The dynamic of the system (conditioned on the number of targets at time t and $t+1$) can, thus, be decomposed as

$$x_{t,i} = F_{\gamma_{t,i}}(x_{t-1,i}, u_{t,i}) \qquad \forall i \in T_t \tag{1}$$

$$\gamma_{t,i} = \gamma_{t-1,i} \qquad \forall i \in T_t. \tag{2}$$

If $i \notin T_{t-1}$ the terms referring to $t-1$ in (1) and (2) correspond to a given prior information. The noise terms $u_{t,i}$ are assumed to be white and pairwise independent.

We assume that the evolution of the number of targets r_t is independent of the previous state of the targets X_{t-1}. The state transition dynamic is, thus, given by (1), (2), and

$$\pi_{r_t, r_{t-1}} = p(r_t|r_{t-1}). \tag{3}$$

We denote by P_b (birth) the probability of switching from $T_t = \{1\}$ to $T_t = \{1, 2\}$, and by P_v (vanishing) the probability of switching from $T_t = \{1, 2\}$ to $T_t = \{1\}$. With these assumptions, we eliminate the problem of the ordering ambiguity found in [6], [7], and [10].

Sensor nodes are usually prone to errors and the measurements available can either arise from the targets of interest when they are detected or be spurious clutter noise (e.g., returns from nearby objects or electromagnetic interferences). One of the

major problems in such a system arises from the generally unknown association between the available measurements and the targets of interest. Traditionally, data association is handled by methods such as the nearest neighbor or the joint probabilistic data association algorithm (JPDA) [3]. When dealing with nonlinear models and unknown number of targets, none of these methods can be applied directly. In this work, a statistical data association scheme is used.

Let m_t be the number and $Z_t = (z_t^1, \ldots, z_t^{m_t})$ be the vector of available measurements at time t. We assume that at most one measurement can arise from each target, and that several measurements can arise from the clutter. The data association vector is denoted by a_t, which is, thus, a vector of length m_t whose components take values in $T_t \cup \{0\}$. Note that $a_t(m) = i$ means that the mth measurement has been generated by the ith target, whereas $a_t(m) = 0$ means that it is a spurious one. Conditioned upon the data association and the state of the system, the measurements are assumed to be independent. The general model for the measurements is, thus, as follows:

$$z_t^m = H_t\left(x_{t,a_t(m)}, v_{t,m}\right), \qquad m \in \{m' | a_t(m') \neq 0\} \quad (4)$$

$$z_t^m \sim p_c(z), \qquad m \in \{m' | a_t(m') = 0\} \quad (5)$$

i.e., a measurement is given by the measurement function H_t (depending on the state of the target and a noise term) if it arises from a target and is specified by some probability distribution p_c if it arises from the clutter. The noise terms $v_{t,m}$ are assumed to be white and pairwise independent.

C. Specific Target Dynamics

In this work, two different motion models were considered: the constant velocity and the coordinated turn rate models. We refer to [9] for an up-to-date survey of motion models. ΔT denotes the length of a time step.

1) Constant Velocity Model: This model is the most commonly used. The target is assumed to move with a constant velocity. For notational simplicity, x_t refers to the state (coordinates and the velocities) of a single target following this motion model and u_t is the corresponding motion noise. We denote with α and β the coordinates: $x_t \triangleq \{\alpha_t, \dot{\alpha}_t, \beta_t, \dot{\beta}_t\}$

$$x_{t+1} = \begin{pmatrix} 1 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 1 \end{pmatrix} x_t + \begin{pmatrix} \frac{\Delta T^2}{2} & 0 \\ \Delta T & 0 \\ 0 & \frac{\Delta T^2}{2} \\ 0 & \Delta T \end{pmatrix} u_t \quad (6)$$

where $u_t \sim \mathcal{N}\left(0, \mathrm{diag}(\sigma_x^2, \sigma_y^2)\right)$.

2) Coordinated Turn Rate Model: This model assumes that the target moves with a constant speed (norm of the velocity vector) and a constant known turn rate ω. Again, we denote x_t as the state of a single target from this class and u_t as the corresponding motion noise. We have

$$x_{t+1} = \begin{pmatrix} 1 & \frac{\sin \omega \Delta T}{\omega} & 0 & -\frac{1-\cos \omega \Delta T}{\omega} \\ 0 & \cos \omega \Delta T & 0 & -\sin \omega \Delta T \\ 0 & -\frac{1-\cos \omega \Delta T}{\omega} & 1 & \frac{\sin \omega \Delta T}{\omega} \\ 0 & \sin \omega \Delta T & 0 & \cos \omega \Delta T \end{pmatrix} x_t + \begin{pmatrix} \frac{\Delta T^2}{2} & 0 \\ \Delta T & 0 \\ 0 & \frac{\Delta T^2}{2} \\ 0 & \Delta T \end{pmatrix} u_t \quad (7)$$

where u_t has the same Gaussian distribution as in (6).

D. Specific Sensing Model

Each sensor provides a set of measurements which can be divided into two distinct sets: the first one consists of the measurements generated by the detected targets and the second one is composed of the false detections from the clutter.

1) Target-Originated Measurements: Many types of sensors provide measurements which are a function of the relative distance between the sensor and the sensed object (e.g., radar, ultrasound, sonar, etc.). Again, we denote by $x_t = \{\alpha_t, \dot{\alpha}_t, \beta_t, \dot{\beta}_t\}$ the state of a single target. The index s will refer to the sensor of interest whose position is $\{\alpha^s, \beta^s\}$. The distance between the sensor and the target is then

$$d_s(x_t) = \left((\alpha_t - \alpha^s)^2 + (\beta_t - \beta^s)^2\right)^{\frac{1}{2}}. \quad (8)$$

A common example is given by measurements of 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 measurements are modeled by

$$h(d) = K - 10\eta \log_{10}(d) \quad (9)$$

$$H_s(x_t, v_t) = h\left(d_s(x_t)\right) + v_t \quad (10)$$

where the measurement noise v_t is assumed to be a zero-mean independent identically distributed (i.i.d.) Gaussian, i.e., $v_t \sim \mathcal{N}(0, R)$; 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 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.

2) Clutter Noise Model: 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 m_t^0 is typically generated by a Poisson distribution with parameter $\lambda V_{\mathrm{meas}}$, where λ is the number of clutter measurements per unit volume and per time step. Hence, we have

$$P\left(m_t^0 = k\right) = e^{-\lambda V_{\mathrm{meas}}} \frac{(\lambda V_{\mathrm{meas}})^k}{k!} \quad (11)$$

$$p_c(z) = \mathcal{U}_{\mathcal{A}_{\mathrm{meas}}}(z). \quad (12)$$

E. Conditional Distribution of the Measurements

The main issue when computing the conditional distribution of a set of measurements resides in the uncertainty in the origin of the measurements. We consider a statistical data association conditioned on the number of targets in the field related to the JPDA method [3]. Because of the arbitrariness in the ordering of the measurements, we assume that (without any knowledge of the value of the measurements) the associations are independent of the current state. The conditional distribution of the measurements can be expanded as

$$p(Z_t | X_t, r_t) = \sum_{a_t} p(Z_t | X_t, r_t, a_t) p(a_t | r_t, m_t). \quad (13)$$

In (13), all possible data associations are enumerated, this is often a major problem when dealing with multiple targets. When a nonlocal sensor is used, the measurements can arise from the

entire field of interest. In order to reduce the complexity of the procedure, it is common to use only a subset of all possible data associations. An association $a_t(m) = i$ would only be allowed if the m^{th} measurement z_t^m is close to the estimate of the expected value $E[H_s(x_{t,i})]$. This idea is referred to as the gating procedure [3]. In our problem, the sensors can only provide local information and, thus, a natural gating is made. The total number of measurements and targets for a leader node will be small and, thus, the number of possible data associations remains small.

The prior probability of a data association (given the number of targets, the number of measurements and the probability of detection P_D) only depends on the set of detected targets and not on the order within the data association vector. The number of data associations in which the same set of targets is detected is given by $\binom{m_t}{m_t^0(a_t)}(m_t - m_t^0(a_t))! = m_t!/(m_t^0(a_t)!)$ (we choose the m_t^0 measurements from the clutter out of the m_t measurements, and then order the $m_t - m_t^0$ measurements from the targets). After some simple calculations, the prior probability of a data association is given by

$$p(a_t|r_t, m_t) \propto P_D^{-m_t^0(a_t)}(1-P_D)^{r_t+m_t^0(a_t))}(\lambda V_{\text{meas}})^{m_t^0(a_t)}. \tag{14}$$

The other term $p(Z_t|X_t, r_t, a_t)$ in (13) is computed by assuming that the measurements are conditionally independent

$$p(Z_t|X_t, r_t, a_t) = V_{\text{meas}}^{-m_t^0(a_t)} \prod_{\{m|a_t(m)\neq 0\}} p\left(z_t^m|x_{t,a_t(m)}\right). \tag{15}$$

Within the framework described above, we aim at performing an on-line estimation of the a posteriori distributions of the target positions, number, and class affiliations $p(X_t|Z_{1:t})$ at time t based on the measurements $Z_{1:t}$ at densely deployed sensor nodes. The exact solution to this problem involves a very high dimensional integration, which is infeasible in practice. We will employ the SMC techniques to solve this problem. The basic principle of SMC is discussed next.

III. Sequential Monte Carlo (SMC) Methods

We consider a generic dynamic model described by

$$\text{initial state model} : p(X_0) \tag{16}$$
$$\text{state transitions model} : p(X_t|X_{t-1}) \quad \forall t \geq 1 \tag{17}$$
$$\text{measurement model} : p(Z_t|X_t) \quad \forall t \geq 1. \tag{18}$$

The cumulative sets of states and measurements are denoted by $X_{0:t} \triangleq (X_0, X_1, \ldots, X_t)$ and $Z_{1:t} \triangleq (Z_1, \ldots, Z_t)$. Suppose an on-line inference of $X_{0:t}$ is of interest. That is, at current time t, we wish to make an estimate of a function of the state variable, say $\psi(X_{0:t})$, based on the currently available observations $Z_{1:t}$. The optimal solution to this problem in the sense of minimum mean-square error is

$$E\{\psi(X_{0:t})|Z_{1:t}\} = \int \psi(X_{0:t})p(X_{0:t}|Z_{1:t})\mathrm{d}X_{0:t}. \tag{19}$$

In most cases, an exact evaluation of this expectation is analytically intractable. SMC methods [11]–[14] are simulation-based techniques, making use of *sequential importance sampling* (SIS), that provide a reliable approximation of this solution.

Let $q(X_{0:t}|Z_{1:t})$ be an *arbitrary* proposal distribution from which we can easily draw samples. Provided that the support of $q(.)$ includes the support of $p(X_{0:t}|Z_{0:t})$, we have the following identity:

$$\begin{aligned} E_p\{\psi(X_{0:t})|Z_{1:t}\} &= \int \psi(X_{0:t})w(X_{0:t})q(X_{0:t}|Z_{1:t})\mathrm{d}X_{0:t} \\ &= E_q\{\psi(X_{0:t})w(X_{0:t})|Z_{1:t}\} \end{aligned} \tag{20}$$

where $w(X_{0:t}) = (p(X_{0:t}|Z_{1:t})/q(X_{0:t}|Z_{1:t}))$ is denoted as the importance weight. Thus, by drawing N random samples $\{X_{0:t}^{(j)}\}_{j=1}^N$ from the proposal distribution $q(X_{0:t}|Z_{1:t})$, it is possible to obtain an estimate of (19) as

$$E_p\{\psi(X_{0:t})|Z_{1:t}\} \simeq \frac{1}{W_t}\sum_{j=1}^N w_t^{(j)}\psi\left(X_{0:t}^{(j)}\right) \tag{21}$$

where $w_t^{(j)} \triangleq w(X_{0:t}^{(j)})$ and $W_t \triangleq \sum_{j=1}^N w_t^{(j)}$. The set, $\{X_{0:t}^{(j)}, w_t^{(j)}\}_{j=1}^N$, of random draws and weights is said to be *properly weighted* with respect to the target distribution $p(X_{0:t}|Z_{1:t})$. One such sample together with its weight is commonly denoted as a *particle*.

A. Sequential Importance Sampling (SIS)

The posterior distribution can be expressed by Bayes' rule as $p(X_{0:t}|Z_{1:t}) = p(Z_{1:t}|X_{0:t})p(X_{0:t})/p(Z_{1:t})$. Therefore, we get the following recursive formula:

$$p(X_{0:t}|Z_{1:t}) = p(X_{0:t-1}|Z_{1:t-1})\frac{p(Z_t|X_t)p(X_t|X_{t-1})}{p(Z_t|Z_{1:t-1})}. \tag{22}$$

This motivates us to adopt a recursive importance sampling strategy by choosing a proposal density which can be factorized as $q(X_{0:t}|Z_{1:t}) = q(X_{0:t-1}|Z_{1:t-1})q(X_t|X_{0:t-1}, Z_{1:t})$. It is then possible to sequentially draw from $q(X_{0:t}|Z_{1:t})$ by keeping the past simulated streams $\{X_{0:t-1}^{(j)}, w_{t-1}^{(j)}\}$ unmodified, and then drawing $X_t^{(j)}$ from $q(X_t|X_{0:t-1}^{(j)}, Z_{1:t})$. The weights in (21) are also recursively updated and become

$$w_t^{(j)} \propto w_{t-1}^{(j)}\frac{p\left(Z_t|X_t^{(j)}\right)p\left(X_t^{(j)}|X_{t-1}^{(j)}\right)}{q\left(X_t^{(j)}|X_{0:t-1}^{(j)}, Z_{1:t}\right)}. \tag{23}$$

B. Resampling Procedure

A common problem with the SIS algorithm is known as the degeneracy phenomenon. In [12], it is shown that the variance of the importance weights can only increase over time which makes the degeneracy problem ineluctable. After a few iterations, some particles will have very small weights. Such samples are said to be ineffective. If there are too many ineffective samples, the Monte Carlo procedure becomes inefficient.

Two options are possible to tackle this problem. The first one involves a good choice for the proposal density (which can be difficult to implement). The second, called *resampling*, is a useful method for reducing ineffective samples and enhancing

effective ones. One simple resampling scheme can be described as follows (cf. [11] for other schemes).

- Draw N sample streams $\{\overline{X}_{0:t}^{(J)}\}_{J=1}^{N}$ from $\{X_{0:t}^{(j)}\}_{j=1}^{N}$ with probabilities proportional to the weights $\{w_t^{(j)}\}_{j=1}^{N}$.
- Assign equal weights to each stream, $\overline{w}_t^{(J)} = N^{-1}$.

It is shown in [15] that samples drawn by the above resampling procedure are indeed properly weighted with respect to $p(X_{0:t}|Z_{1:t})$, provided that N is sufficiently large.

The degeneracy of the particles can be measured by the effective sample size N_{eff} defined as

$$N_{\text{eff}} \triangleq N. \left[1 + \text{Var}\left(\frac{p\left(X_t^{(j)}|Z_{1:t}\right)}{q\left(X_t^{(j)}|X_{0:t-1}^{(j)}, Z_{1:t}\right)}\right)\right]^{-1}. \quad (24)$$

It can be approximated by $\widehat{N_{\text{eff}}} = (\sum_{j=1}^{N}(w_t^{(j)})^2)^{-1}$ [14]. Heuristically, $\widehat{N_{\text{eff}}}$ reflects the equivalent size of a set of i.i.d. samples for the set of N weighted ones. It is suggested in [11] and [14] that resampling should be performed whenever the effective sample size becomes small, e.g., $\widehat{N_{\text{eff}}} \leq (N/10)$.

IV. Joint Single Target Tracking and Classification

The class of a target is an important information as such. Furthermore, when tracking a target whose maneuvering capabilities are unknown, the use of a very general motion model can lead to very poor estimates. This uncertainty is often due to the lack of knowledge about the *type* of the tracked object and only a finite set of types is of interest. Another approach when dealing with such an uncertainty is to compare several classes of dynamic models such as those presented in Section II. In this section, we propose an algorithm for jointly tracking and classifying a single target evolving within a sensor network. A class-based resampling scheme is developed.

A. Related Work

In [16], the authors introduced a Bayesian target classification method based on the estimate of kinematics only. Their major contribution was to point out the dependence between the target state and the target class, and then to integrate this dependence into a joint tracking and classification algorithm. Within this framework, the estimations are provided by a grid-based algorithm, which is known to be very difficult to implement, especially in high dimensional spaces.

In [17], the multitarget tracking and classification problem is addressed (for a fixed number of targets). Their implementation is based on a nearest-neighbor data association. Moreover, the class estimate always settles on a fixed value. This is problematic when two classes appear similar during a certain period of time, since the class estimation might lock on the wrong class. To solve this problem, it is proposed in [18] (for a single target) to use a separate particle filter for each possible class and a method for comparing different filters is given.

The problem of joint tracking and classification can be seen as that of simultaneously dealing with both a fixed model parameter (class) and state variables (position and velocity). Several works have proposed algorithms for dealing with static parameters within an SMC framework [19], [20]. However, the parameters considered were continuous which is fundamentally different from our classification problem.

Our approach combines the advantages of the previous works. We use the SMC framework, but we make sure to always be able to recover information related to a specific class, and in order to devote more (but not all) computational load to the more likely classes, our approach will compound the different class information into a single filter.

B. Class-Based Resampling Scheme

Our algorithm relies on the framework presented in Section II but for the sake of clarity, here, we focus on the case of a single target. Extension to the multiple target case will be presented in Section V. In order to simplify the notations, we will here consider $X_t = (x_t, \gamma)$ as the joint state and class of the target of interest.

The simplest way of dealing with the class is to include it in the state vector and then to use a simple particle filter for this augmented state. This makes use of the static evolution model (2) of the class parameter. Because of this absence of dynamic, the number of particles from each class will not change during the updating step. A change will only occur during the resampling stage. Sometimes, and especially during the initial steps, the particle filter may lock on the wrong class. This situation can sometimes last for quite a while. Then, all particles might eventually settle in a wrong class. To avoid this situation, one idea is to assume an artificial evolution (usually a Markov chain) of the class parameter (2). This results in a model mismatch and, because only the recent observations still have an influence on the class estimation, this also leads to a *loss of information* as argued in [20].

Because of the finite number of available classes, it is possible to keep particles for each of those. If we assume that a sufficient number of particles from each class remains available at each time, then it would always be possible to recover from a misclassification. Subsequently, our aim is to keep a sufficient number of particles per class. Since a change in the number of particles arise only during the resampling, this suggests modifying the resampling algorithm.

1) The Algorithm: The resampling scheme is essentially a way of eliminating trajectories with small weights and amplifying those with large weights. This can be applied within the set of particles belonging to the same class. It is indeed clear that the usual resampling scheme can be done by first setting the number of particles N_γ for each class according to a multinomial distribution with parameters the posterior class probabilities, and then by drawing N_γ streams within the particles with class γ according to their weights. Here, we introduce a threshold step to keep a sufficient number of particles for each class and keep a constant total number of particles. Due to the threshold operation, all particles will not be assigned equal weights. The resampling scheme should not change the class probabilities and, thus, the total weight corresponding to a given class should not change. Furthermore, within a class, we draw the indexes according to the weights, all particles within a class should then have equal weights. Similar ideas can be found in the stratified sampling theory and future work on these connections will certainly be profitable.

We denote Λ as the set of all possible classes. The class-based resampling scheme can be summarized as follows.

- Choose the number of particles for each class N_γ according to a multinomial distribution with parameters $\{\hat{P}(\gamma|Z_{0:t})\}_{\gamma\in\Lambda}$.
- $\forall\gamma\in\Lambda$, if $N_\gamma < N_{\text{Threshold}}$, set $N_\gamma = N_{\text{Threshold}}$.
- Reduce the number of particles from the class with the most particles until $\sum_{\gamma\in\Lambda} N_\gamma = N$.
- $\forall\gamma\in\Lambda$, draw N_γ sample streams $\{\overline{X}_{0:t}^{(J)}\}$ with probability proportional to the weights $\{w_t^{(j)}|j\in\{j'|\gamma_t^{(j')}=\gamma\}\}$ from $\{X_{0:t}^{(j)}|j\in\{j'|\gamma_t^{(j')}=\gamma\}\}$.
- Assign equal weight to each new sample within a class i.e. $\overline{w}_t^{(J)} = (\hat{P}(\gamma^{(J)}|Z_{0:t})/N_{\gamma^{(J)}})$.

For a sufficiently large $N_{\text{Threshold}}$, it is possible to show [15] that if the set of samples representing $P(x_t|\gamma, Z_{0:t})$ is properly weighted, then it is also properly weighted after resampling within a class. Therefore, by using the expansion $P(x_t|Z_{0:t}) = \sum_\gamma P(x_t|\gamma, Z_{0:t})P(\gamma|Z_{0:t})$ our complete set of samples will remain properly weighted with respect to $P(x_t|Z_{0:t})$. Just as no rigorous scheme exists in the literature to select the optimal number of particles, $N_{\text{Threshold}}$ remains a tuning parameter.

V. Multiple Target Tracking and Classification

As stated in Section II, we make use of the SMC techniques within a leader-based tracking scheme. In this framework, it is possible to consider each target *independently*, as long as the other one are far away. It is subsequently compulsory to deal with a varying number of targets. Indeed, as we mentioned earlier, a target perturbing the measurements available to a sensor cannot be considered as noise. In this section, we will describe the main body of the general algorithm.

A. Related Work

In order to deal with an unknown or varying number of targets r_t, several alternatives are available. A classical approach is to estimate r_t separately from the rest of the state-space by using a hypothesis test for instance, and then to treat the estimated r_t as the true number of targets for the estimation of the other variables [4]. Another possibility is to compare several tracking hypotheses with different number of targets. In [10], random sets and finite-set statistics are employed to achieve this objective. In order to estimate the state of the system, it is then necessary to find the peaks in the probability hypothesis density (the equivalent of the probability density function for random sets) using, e.g., the EM algorithm. In [6] and [7], it is proposed to cast the multiple target tracking problem into that of filtering a JMS, where the number of targets and, thus, the dimensionality of the state, follows a Markov chain.

Our approach also makes use of the JMS in modeling the varying number of targets. However, in order to meet the requirements of the sensor networks, we will focus on maintaining the computational complexity as low as possible. We will also incorporate the class-based resampling scheme described in Section IV so as to tackle the issue of jointly tracking and classifying the targets.

B. SMC Solution

Our state becomes (X_t, r_t) with $X_t = \{x_{t,i}, \gamma_i\}_{i\in T_t}$, and where we recall that T_t is the set of active targets. Our goal remains to sequentially estimate $p(X_t|Z_{1:t})$. Because we are given a dynamic model, the SMC methods presented in Section III are well fit to solve this problem.

Optimal Sampling Density: The first issue encountered in this approach resides in the choice of the sampling density. In fact, the optimal choice for the sampling density is $q(X_t, r_t|X_{0:t-1}^{(j)}, r_{0:t-1}^{(j)}, Z_{1:t}) = p(X_t, r_t|X_{t-1}^{(j)}, r_{t-1}^{(j)}, Z_t)$ [12]. Clearly, it is impossible to sample directly from this distribution and even if we could, the weight update would also require the evaluation of $p(Z_t|X_{t-1}^{(j)})$, which does not admit a closed-form expression.

For the above reasons, using the prior distribution as the sampling density is often a reasonable choice (for instance, in the single target scenario). When dealing with r_t, the problem is more subtle. If the probability of a new target appearance is small, only a few particles will increase their dimension and, thus, become accurate, when a new target enters the field. After the resampling stage, only a few number of particles will be kept, which will result in a loss of diversity within the particles and could lead to loosing track of the target of interest. On the other hand, assuming a large probability of appearance would lead to drawing many inaccurate samples and overestimating the probability of having a second target in the field. It is, therefore, imperative to include the current observation in the proposal distribution of r_t. The optimal sampling density can be written as

$$q\left(X_t, r_t|X_{0:t-1}^{(j)}, r_{0:t-1}^{(j)}, Z_{1:t}\right) = p\left(X_t|r_t, X_{t-1}^{(j)}, r_{t-1}^{(j)}, Z_t\right) p\left(r_t|X_{t-1}^{(j)}, r_{t-1}^{(j)}, Z_t\right) \quad (25)$$

where $p(r_t|X_{t-1}^{(j)}, r_{t-1}^{(j)}, Z_t) \propto p(Z_t|X_{t-1}^{(j)}, r_{t-1}^{(j)}, r_t)p(r_t|r_{t-1}^{(j)})$. The first term expands as

$$\begin{aligned} p\Big(Z_t|X_{t-1}^{(j)}, r_{t-1}^{(j)}, r_t\Big) &= \sum_{a_t} p\Big(Z_t|X_{t-1}^{(j)}, a_t, r_t\Big)p(a_t|r_t, m_t) \\ &= \sum_{a_t} p(a_t|r_t, m_t)V_{meas}^{-m_t^0(a_t)} \\ &\quad \times \prod_{\{m|a_t(m)\neq 0\}} p\Big(z_t^m|x_{t-1,a_t(m)}^{(j)}\Big). \end{aligned} \quad (26)$$

Finally, the computation of (26) requires

$$p\left(z_t^m|x_{t-1,a_t(m)}^{(j)}\right) = \int p\left(z_t^m|x_{t,a_t(m)}\right) \times p\left(x_{t,a_t(m)}|x_{t-1,a_t(m)}^{(j)}\right) \mathrm{d}x_{t,a_t(m)}. \quad (27)$$

Choice of the Sampling Density: The quantity in (27) can be approximated by an unscented transform [21], as proposed in [6] and [7], or simpler by using the mean or mode of this distribution. This choice is also used in the auxiliary particle filter to compute the auxiliary weights [22]. In our simulations, this approximation appeared sufficient for the previously existing tar-

gets. Since $F_\gamma(x,u)$ is linear in u and u is zero-mean, our approximation of (27) is $\forall i \in T_{t-1} \cap T_t$

$$p\left(z_t^m | x_{t-1,i}^{(j)}, a_t(m)=i\right) \simeq p\left(z_t^m | \mu_{t,i}^{(j)}\right) \tag{28}$$

$$\mu_{t,i}^{(j)} = E\left[x_{t,i} | x_{t-1,i}^{(j)}, \gamma_i\right] = F_{\gamma_{t,i}}(x_{t-1,i}, 0). \tag{29}$$

For the newly appeared targets, there is no available $x_{t-1,i}^{(j)}$, and in that case, (28) is not accurate. Therefore, we need to compute $p(z_t^m | a_t(m) \in T_t \cap \overline{T_{t-1}})$ (i.e., (27) for new targets). If no prior information on the position of the newly appeared target is available, it is common to consider $p(x_{t-1,i} | a_t(m) \in T_t \cap \overline{T_{t-1}})$ as uniform in the sensed volume. When the sensing model is given by (4), $p(z_t^m | a_t(m) \in T_t \cap \overline{T_{t-1}})$ can then easily be computed analytically. (It is an exponential of $-z_t^m$ times a difference between two Gaussian cumulative distribution functions (CDFs) omitted here for the sake of brevity). Alternatively, if a specific prior on the position is used, it is possible to sample N particles for the newly appeared target, and approximate $p(z_t^m | a_t(m) \in T_t \cap \overline{T_{t-1}})$ by Monte Carlo integration. The same samples can afterwards be used when drawing $X_{t+1}^{(j)}$. We now have an approximation of (26) denoted as $\hat{p}_\mu(Z_t | X_{t-1}^{(j)}, r_{t-1}^{(j)}, r_t)$.

In (25), $p(X_t | r_t, X_{t-1}^{(j)}, Z_t) = \prod_{i \in T_t} p(X_{t,i} | X_{t-1,i}^{(j)}, Z_t)$ could be approximated by performing a local linearization of the dynamic model similar to the extended Kalman filter [12], [14], but this would result in a heavier computational load and, thus, consumes more power, whereas sampling from the prior is sufficient once the number of targets is given. Nevertheless, for the newly appeared targets, sampling from the prior distribution can be inefficient if we assume a uniform distribution of the position within the entire area covered by the leader node. It is, however, possible here to get an analytical expression of $p(X_{t,i} | i \in T_t \cap \overline{T_{t-1}}, Z_t)$. It is easily shown that with our choice of $H_t, p(d_s(x_{t,i})^2 | i \in T_t \cap \overline{T_{t-1}}, z_t^m)$ follows a lognormal distribution (omitted here for the sake of brevity).

Subsequently, the following proposal density is used:

$$q\left(X_t, r_t | X_{t-1}^{(j)}, r_{t-1}^{(j)}, Z_t\right) \propto \left(p\left(r_t | r_{t-1}^{(j)}\right) \hat{p}_\mu\left(Z_t | X_{t-1}^{(j)}, r_t\right)\right) \times \prod_{i \in T_t \cap T_{t-1}} p\left(X_{t,i} | X_{t-1,i}^{(j)}\right) \prod_{i \in T_t \cap \overline{T_{t-1}}} p(X_{t,i} | Z_t). \tag{30}$$

Extension of the Class-Based Resampling Scheme: In Section IV, we proposed a particle filter approach for dealing with the classification. As the number of targets is now greater than one, a direct generalization of the proposed algorithm would be to keep a sufficient number of particles per class association vector. However, this would result in a substantial increase in the minimum number of particles we should keep. Since we use a leader-based tracking scheme, our attention is mainly focused on the first target. Hence, for the other targets, we chose to depart from the real static evolution of the class and allow a small probability of switching between classes so as not to settle in the wrong class. The robust classification for those targets is actually made by the leader node responsible for them.

VI. Sensor Selection

Because our tracking scheme relies on a leader-based algorithm, the sensor selection step is essential. In this section, we consider an information driven sensor selection algorithm. The choice of the sensor will determine the efficiency of the tracking (and, thus, of the classification) and the resource consumption of the nodes (e.g., power use). Depending on the cost of a handoff to different node, it could be necessary to penalize such an operation. A tradeoff is typically to be made between the information gain and the total cost [5]. A simple formulation of the sensor selection scheme can be given as that of maximizing the expectation of an objective function. We denote by s_t the leader node selected at time t and emphasize the dependence of the various densities on this variable. This selection can be written as

$$s_{t+1} = \arg\max_s \left(E_{p_s(Z_{t+1}|Z_{1:t})}\left[\alpha \Upsilon_{\mathrm{u}}(s) + (1-\alpha)\Upsilon_{\mathrm{c}}(s)\right]\right) \tag{31}$$

where Υ_{u} is the information gain by fusion of a set of measurements from the sensor s, Υ_{c} is the cost of choosing the sensor, and α is the relative weight between those quantities.

The function Υ_{c} is characterized by link bandwidth, transmission latency, node battery power reserve, etc. In our case, this is the cost of handing the current belief state off to sensor s, acquiring data at sensor s, and combining the data with the current belief. This quantity is typically deterministic (e.g., function of the distance between the sensors) and the selection criterion becomes an immediate extension of the most informative selection scheme. In this section, we will, therefore, focus on the expected information gain $E_{p_s(Z_{t+1}|Z_{1:t})}[\Upsilon_{\mathrm{u}}(s)]$. As evidenced by the presence of the expectation, such a selection criterion is based on the current belief only and does not use any new measurements.

In order to measure the information gain in choosing sensor s, we will make use of the notion of mutual information, which is a common criterion to measure the reduction of uncertainty in a random variable due to the knowledge of another one [23]. This criterion is also advocated in [2], where its approximation relies on a grid-based method. Another commonly used criterion for sensor management is the α-divergence (or Renyi divergence) [24], which reduces to our choice as $\alpha \to 1$. Our algorithm can be straightforwardly extended to use the α-divergence. In [25], the author deals the issue of sensor management when the multitarget problem is addressed within the finite-set statistics.

Let $U \in \mathcal{U}$, $V \in \mathcal{V}$, and $W \in \mathcal{W}$ be random variables having a conditional density $p(u,v|w)$. Conditioned on a single realization w of W (and not on the random variable W), the mutual information between U and V is given by

$$I(U;V|W=w) \triangleq E_{p(u,v|w)}\left[\log \frac{p(u,v|w)}{p(u|w)p(v|w)}\right]. \tag{32}$$

Let p_1 and p_2 be two probability densities, the Kullback–Leibler (KL) divergence between p_1 and p_2 is defined by

$$D(p_1 \| p_2) \triangleq E_{p_1}\left[\log \frac{p_1}{p_2}\right]. \tag{33}$$

Expected Information Gain: As mentioned earlier, within our tracking framework, we are mainly concerned about the

first target of each leader node. We will consequently consider the information conveyed about this target only. We have

$$s_{t+1} = \arg\max_{s} I_s(x_{t+1,1}; \tilde{z}_{t+1}|Z_{1:t}) \tag{34}$$

where the conditioning is on the observed realization of $Z_{1:t}$ and the sensors s of interest are in a specified neighborhood of s_t (e.g., the three closest sensors to the predicted position of the first target). $\tilde{z}_{t+1}$ is a random variable denoting the measurement at sensor s that would arise from the first target. There is indeed no need to consider a data association problem.

From the definition (32) of the mutual information and after some simple calculations, we have

$$\begin{aligned} E[\Upsilon_{\mathrm{u}}(s)] &= I_s(x_{t+1,1}; \tilde{z}_{t+1}|Z_{1:t}) \\ &= E_{p_s(x_{t+1,1},\tilde{z}_{t+1}|Z_{1:t})}\left[\log \frac{p_s(x_{t+1,1}, \tilde{z}_{t+1}|Z_{1:t})}{p(x_{t+1,1}|Z_{1:t})p_s(\tilde{z}_{t+1}|Z_{1:t})}\right] \\ &= E_{p_s(\tilde{z}_{t+1}|Z_{1:t})} \\ &\quad \times \left[D\left(p_s(x_{t+1,1}|Z_{1:t}, \tilde{z}_{t+1})\|p(x_{t+1,1}|Z_{1:t})\right)\right]. \end{aligned} \tag{35}$$

Our selection criterion can, thus, also be seen as that of maximizing the average KL distance between the one-step ahead filtering density and the predictive density. The latter being performed with respect to the predictive density of the measurements.

This quantity will be computed as proposed in [6] by a Monte Carlo method. From (32), we get

$$\begin{aligned} E[\Upsilon_{\mathrm{u}}(s)] = E_{p_s(\tilde{z}_{t+1}|x_{t+1,1})p_s(x_{t+1,1}|Z_{1:t})} \\ \times \left[\log \frac{p_s(\tilde{z}_{t+1}|x_{t+1,1})}{p_s(\tilde{z}_{t+1}|Z_{1:t})}\right]. \end{aligned} \tag{36}$$

In order to estimate (36), it is possible to use N samples $\{(x_{t+1,1}^{(j)}, \tilde{z}_{t+1}^{(j)})\}_{j=1}^{N}$ properly weighted with respect to $p_s(\tilde{z}_{t+1}|x_{t+1,1})p(x_{t+1,1}|Z_{1:t})$ and get

$$E[\Upsilon_{\mathrm{u}}(s)] \simeq \frac{1}{N}\sum_{j=1}^{N} \log \frac{p_s\left(\tilde{z}_{t+1}^{(j)}|x_{t+1,1}^{(j)}\right)}{p_s\left(\tilde{z}_{t+1}^{(j)}|Z_{1:t}\right)}. \tag{37}$$

We will first approximate $p(x_{t+1,1}|Z_{1:t})$. Using the trial distribution $q(x_{t+1,1}) = p(x_{t+1,1}|x_{t,1}^{(j)})$, we can draw samples $\{x_{t+1,1}^{(j)}, w_{t+1}^{(j)})\}_{j=1}^{N}$ with the importance weight given by $w_{t+1}^{(j)} = p(x_{t+1,1}^{(j)}|Z_{1:t})/q(x_{t+1,1}^{(j)}|Z_{1:t}) = w_t^{(j)}$. Based on the importance sampling principle, $\{x_{t+1,1}^{(j)}, w_{t+1}^{(j)})\}_{j=1}^{N}$ is easily shown to be properly weighted with respect to the distribution $p(x_{t+1,1}|Z_{1:t})$. These samples are not *sensor dependent* and can, thus, be kept for all considered sensors. We can now sample $\tilde{z}_{t+1}^{(j)} \sim p_s(\tilde{z}_{t+1}|x_{t+1,1}^{(j)})$. By composition it can also be shown that the set of samples and weights, $\{(x_{t+1,1}, \tilde{z}_{t+1})^{(j)}, w_{t+1}^{(j)})\}_{j=1}^{N}$, is properly weighted with respect to $p_s(x_{t+1,1}, \tilde{z}_{t+1}|Z_{1:t})$. We now need to approximate $p_s(\tilde{z}_{t+1}^{(j)}|Z_{1:t})$, which is not directly available with those samples. However, by expanding this term as

$$p_s(\tilde{z}_{t+1}|Z_{1:t}) = \int p_s(\tilde{z}_{t+1}|x_{t+1,1})p(x_{t+1,1}|Z_{1:t})\mathrm{d}x_{t+1,1} \tag{38}$$

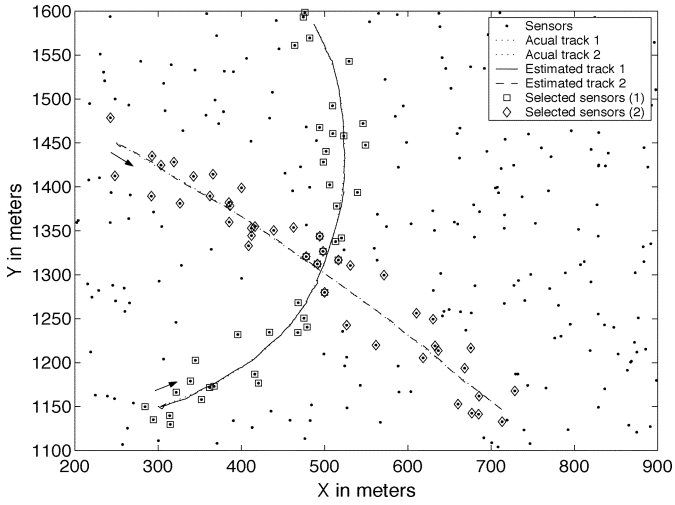


Fig. 1. Actual and estimated trajectories for two targets, one sample run.

and using the set of samples and weights $\{x_{t+1,1}^{(j)}, w_{t+1}^{(j)})\}_{j=1}^{N}$, we get the following approximation:

$$\hat{p}_s\left(\tilde{z}_{t+1}^{(j)}|Z_{1:t}\right) = \sum_{k=1}^{N} w_{t+1}^{(k)} p_s\left(\tilde{z}_{t+1}^{(j)}|x_{t+1,1}^{(k)}\right). \tag{39}$$

Finally, the expected information gain can be approximated by

$$E[\Upsilon_{\mathrm{u}}(s)] \simeq \sum_{j=1}^{N} w_{t+1}^{(j)} \log \frac{p_s\left(\tilde{z}_{t+1}^{(j)}|x_{t+1,1}^{(j)}\right)}{\hat{p}_s\left(\tilde{z}_{t+1}|Z_{1:t}^{(j)}\right)}. \tag{40}$$

The complexity of this scheme is $\mathcal{O}(N^2)$ which can appear computationally intensive. Therefore, solutions might be needed to reduce its complexity. Our goal here is not to approximate accurately the aforementioned expectation but only to find the index s maximizing it. Hence, a rough approximation should be sufficient. One option would be to select P (with $P \ll N$) initial samples by a usual resampling stage, and then perform this scheme with equal weights. We could also think of inserting this step every k steps or dynamically performing this scheme when, for instance, the Monte Carlo variance of the particles gets above a given threshold. Another issue in the sensor selection is that a given sensor can already be assigned a task. In such a case, the belief would be handed off to the second best node and so on.

VII. Simulation Results

To illustrate the performance of the proposed algorithm, simulations are performed. The scenario under consideration presents a crossing of two targets from different classes. Both targets are given a leader node at time $t = 0$. Fig. 1 gives a picture of the considered scenario. The first target (starting at the lower left corner of Fig. 1) is chosen to belong to the second class, i.e., the coordinated turn rate model (7)), while the second target (starting at the upper left corner of Fig. 1) belongs to the first class, i.e., the constant velocity model (6). The true trajectories of both targets are represented in Fig. 1 with dotted lines. The arrows indicate the direction of the motion.

The parameters of the simulations are shown in Table I. For the second target of each leader node, the probability of

TABLE I
PARAMETERS OF THE SIMULATIONS

$P_D = 0.95$ if in range	$P_D = 0$ otherwise
$K = 9\text{dBm}$	$\eta = 3$
Sensing range: $[4\text{m}, 50\text{m}]$	Meas. range: $[-41\text{dBm}, -9\text{dBm}]$
$\lambda V_{meas} = 0.2$	$V_{meas} = 33\text{dBm}$
500 particles per leader	80 particles per class minimum
$P_b = 0.005$	$P_v = 0.01$
CV model: $\sigma_x = \sigma_y = 0.005$	CTR model $\sigma_x = \sigma_y = 0.006$
$T = 1$ second	$R = \frac{1}{2}$
Field: 700×500 m^2	300 randomly scattered nodes

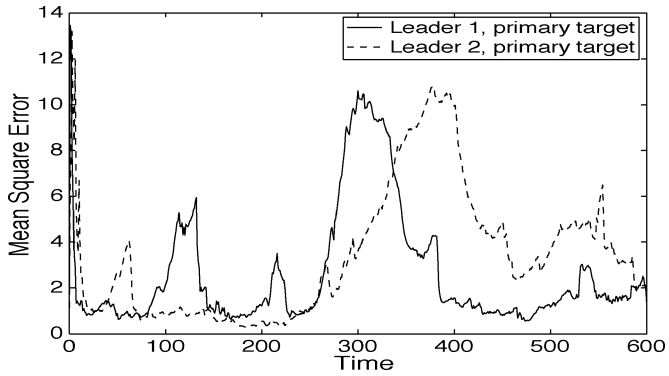


Fig. 2. Mean-square error on the position for 20 runs.

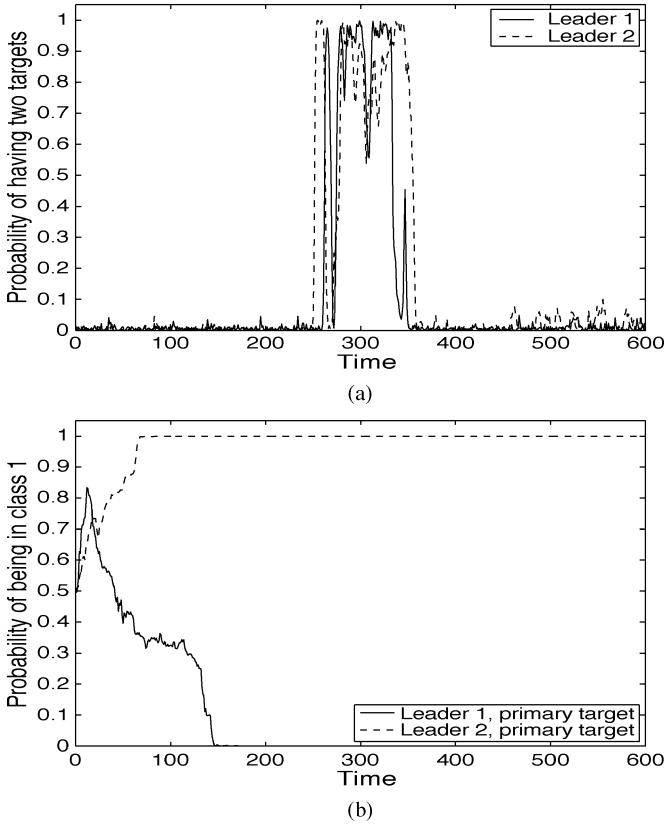


Fig. 3. (a) Probabilities of having two targets in the field. (b) Probabilities that the first target of the leader node is from the first class. 20 runs average.

changing the class is chosen as 0.01. The prior for the target class is assumed to be uniform. The prior for the initial state is Gaussian with the true mean and the covariance matrix $\Sigma = \text{diag}(10^2, 10^2, 0.005^2, 0.005^2)$. For the possibly appearing target, the prior on the position is taken as uniform in the field cover by a leader node, while the prior for the velocity is taken as Gaussian with the mean and variance of the true velocities of the other target on the entire track. We, therefore, assume that we have a very limited information about the other target. Indeed our nodes only have observation related to the distance of the objects which makes the initialization a very hard problem without any *a priori* knowledge. This information on the velocity could for example be transmitted once in a while to a broad neighborhood of nodes.

To evaluate the performance of the proposed algorithm, we have performed 20 consecutive runs of the scenario. The true trajectories are kept identical to those shown in Fig. 1 but independent measurements are simulated for each run. We show in Fig. 2 the mean-square error on the position over those 20 runs. Our algorithm is able to accurately track each of the targets. The mean width of the three sigma ellipsoid on the position is 5.08 and 5.01 m for the primary target of the first- and second-leader node, respectively.

The performance of the classification and target detection are shown in Fig. 3. We can see that for both leader nodes, the probability of having two targets in the field jumps as the target enters the field. It does not happen at the same instant for the two leader nodes because it depends on the specific chosen node. For the classification, we can see that during the initial steps, the first-leader node misclassifies its target, but is able to recover and correctly classifies the target after some time. This shows the ability to recover from a misclassification.

VIII. CONCLUSION

In this paper, we have considered the application of the SMC methodology to the problem of jointly tracking several targets and identifying them down to a specific class related to their motion model. The scenario under consideration is a collaborative sensor network, where a nonlinear sensing model is assumed. The requirements of such networks (e.g., low-power consumption, distributed processing, etc.) have led us to the use of a scalable leader-based scheme so as to solve this complex problem through collaboration among the sensors and by dividing the task into several easier, localized ones. Each leader node faces the simplest nontrivial multitarget scenario. The computational burden of multitarget tracking is, thus, mitigated. It is reasonable to imagine that future sensor networks (in which communication is the bottleneck as opposed to computation) will be able to cope with SMC filtering in this setting. Our first contribution is the extension of the leader-based tracking scheme to a multitarget case by using a low computational complexity version of the filtering of JMSs to deal with the varying number of targets. The second contribution resides in the design of a class-based resampling scheme leading to a robust classification of the targets, while allowing more computational load where needed. Finally, we have presented an SMC method to solve the problem of information-driven sensor selection. Our algorithm is able to

detect an appearing target and simulations have shown that the targets are tracked and classified accurately.

REFERENCES

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

[2] J. Liu, J. Reich, and F. Zhao, "Collaborative in-network processing for target tracking," *EURASIP J. Appl. Signal. Process.*, no. 4, pp. 378–391, Mar. 2003.

[3] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. New York: Academic, 1988.

[4] C. Hue, J.-P. L. Cadre, and P. Perez, "Sequential Monte Carlo methods for multiple target tracking and data fusion," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 309–325, Feb. 2002.

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

[6] A. Doucet, B. Vo, C. Andrieu, and M. Davy, "Particle filtering for multitarget tracking and sensor management," in *Proc. Int. Conf. Inform. Fusion*, vol. 1, 2002, pp. 474–481.

[7] C. Andrieu, M. Davy, and A. Doucet, "Efficient particle filtering for jump Markov systems: Application to time-varying autoregressions," *IEEE Trans. Signal Process.*, vol. 51, no. 7, pp. 1762–1770, Jul. 2003.

[8] J. Liu, J. Liu, J. Reich, P. Cheung, and F. Zhao, "Distributed group management for track initiation and maintenance in target localization applications," in *Proc. Inf. Process. Sensor Netwo.: Second Int. Workshop, IPSN03*, vol. 2634, F. Zhao and L. Guibas, Eds., Palo Alto, CA, Apr. 2003, pp. 113–128.

[9] X. Li and V. Jilkov, "A survey of maneuvering target tracking: Dynamic models," in *Proc. SPIE: Signal Data Process. Small Targets 2000*, vol. 4048-22, Apr. 2000, pp. 212–235.

[10] H. Sidenbladh and S. Wirkander, "Particle filtering for random sets," *IEEE Trans. Aerosp. Electron. Syst.*, to be published.

[11] J. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *J. Amer. Statist. Assoc.*, vol. 93, no. 443, pp. 1032–1044, 1998.

[12] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statist. Comput.*, vol. 10, no. 3, pp. 197–208, 2000.

[13] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2001.

[14] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[15] D. Crisan and A. Doucet, "Convergence of sequential Monte Carlo methods," CUED-F-INFENG, Tech. Rep. 381, 2000.

[16] S. Challa and G. W. Pulford, "Joint target tracking and classification using radar and ESM sensors," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 37, no. 3, pp. 1039–1055, Jul. 2001.

[17] S. Herman and P. Moulin, "A particle filtering approach to FM-band passive radar tracking and automatic target recognition," in *Proc. IEEE Aerosp. Conf.*, vol. 4, 2002, pp. 1789–1808.

[18] N. Gordon, S. Maskell, and T. Kirubarajan, "Efficient particle filters for joint tracking and classification," in *Proc. SPIE: Signal Data Process. Small Targets*, vol. 4728, 2002, pp. 439–449.

[19] G. Storvik, "Particle filters for state-space models with the presence of unknown static parameters," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 281–289, Feb. 2002.

[20] J. Liu and M. West, "Combined parameter and state estimation in simulation-based filtering," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds. New York: Springer-Verlag, 2001.

[21] S. J. Julier and J. K. Uhlmann, "The scaled unscented transformation," in *Proc. IEEE Amer. Control Conf.*, May 2002, pp. 4555–4559.

[22] M. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *J. Amer. Statist. Assoc.*, vol. 94, no. 446, pp. 590–599, 1999.

[23] T. Cover and J. Thomas, *Elements of Information Theory*. New York: Wiley, 1990.

[24] C. Kreucher, K. Kastella, and A. Hero, "A Bayesian method for integrated multitarget tracking and sensor management," in *Proc. IEEE Inform. Fusion*, vol. 1, 2003, pp. 704–711.

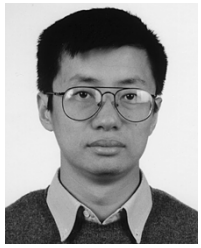
[25] R. Mahler, "Objective functions for Bayesian control-theoretic sensor management, 1: Multitarget first-moment approximation," in *Proc. IEEE Aerosp. Conf.*, vol. 4, Mar. 2003, pp. 1905–1923.

Tom Vercauteren (S'05) received the Ingénieur degree from Ecole Polytechnique, Paris, France, in 2003 and the M.S. degree in electrical engineering from Columbia University, New York, in 2004. He is now working towards the Ph.D. degree from INRIA, Sophia-Antipolis, France.

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

Dong Guo received the B.S. degree in geophysics and computer science from the China University of Mining and Technology (CUMT), Xuzhou, China, in 1993, the M.S. degree in geophysics from the Research Institute of Petroleum Exploration and Development (RIPED), Beijing, China, in 1996, the Ph.D. degree in applied mathematics from Beijing University, Beijing, China, in 1999, and the second Ph.D. degree in electrical engineering from Columbia University, New York, in 2004.

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



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 in the Department of Electrical Engineering, Texas A&M University, College Station. In January 2002, he joined the Department of Electrical Engineering, Columbia University, New York, where he is currently an Associate Professor. Among his publications is a recent book entitled *Wireless Communication Systems: Advanced Techniques for Signal Reception* (Englewood Cliffs: Prentice-Hall, 2003). 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. He has published extensively in these areas. His current research focuses include Bayesian Monte Carlo signal processing, wireless communications, and genomic signal processing.

Dr. Wang received the 1999 National Science Foundation (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 TRANSACTIONS ON COMMUNICATIONS, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and the IEEE TRANSACTIONS ON INFORMATION THEORY.