

## Research Article

# An Information-Theoretic Approach for Energy-Efficient Collaborative Tracking in Wireless Sensor Networks

**Loredana Arienzo**

*Institute for the Protection and Security of the Citizen, Joint Research Centre, European Commission, Ispra, 21027 Varese, Italy*

Correspondence should be addressed to Loredana Arienzo, loredana.arienzo@jrc.it

Received 3 November 2009; Accepted 24 February 2010

Academic Editor: Xinbing Wang

Copyright © 2010 Loredana Arienzo. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The problem of collaborative tracking of mobile nodes in wireless sensor networks is addressed. By using a novel metric derived from the energy model in LEACH (W.B. Heinzelman, A.P. Chandrakasan and H. Balakrishnan, Energy-Efficient Communication Protocol for Wireless Microsensor Networks, in: Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS '00), 2000) and aiming at an efficient resource solution, the approach adopts a strategy of combining target tracking with node selection procedures in order to select informative sensors to minimize the energy consumption of the tracking task. We layout a cluster-based architecture to address the limitations in computational power, battery capacity and communication capacities of the sensor devices. The computation of the posterior Cramer-Rao bound (PCRB) based on received signal strength measurements has been considered. To track mobile nodes two particle filters are used: the bootstrap particle filter and the unscented particle filter, both in the centralized and in the distributed manner. Their performances are compared with the theoretical lower bound PCRB. To save energy, a node selection procedure based on greedy algorithms is proposed. The node selection problem is formulated as a cross-layer optimization problem and it is solved using greedy algorithms.

## 1. Introduction

Wireless sensor networks (WSNs), which normally consist of hundreds or thousands of sensor nodes each capable of sensing, processing, and transmitting environmental information, are deployed to monitor certain physical phenomena or to detect and track certain targets in an area of interests [1]. The issue of tracking moving targets in WSNs has received significant attention in recent years [2–4]. As any other algorithm designed for sensor networks, a tracking algorithm should be: (1) self-organizing, i.e. it should not depend on global infrastructure; (2) energy efficient, i.e. it should require little computation and, especially, communication; (3) robust, i.e. it should not depend on noise and movement of the target; (4) accurate, i.e. it should work with accuracy and precision in various environments, and should not depend on sensor-to-sensor connectivity in the network; (5) reliable, i.e. it should be tolerant to node failures.

Tracking a target in sensor networks is mainly challenging regardless of the energy consumption due to resource-constrained features of such a network. The minimization of energy consumption for a sensor network with target activities is complicated since target estimation involves collaborative sensing and communication between different nodes. The problem of selecting the best nodes for tracking a target in a distributed wireless sensor network was investigated since [5].

The main idea is for a network to determine participants in a *sensor collaboration* by dynamically optimizing the utility function of data for a given cost of communication and computation.

Previous research [6–9] has focused on information-theoretic node selection approaches, that is, on heuristics to select an informative sensor such that the fusion of the selected sensor observation with the prior target location distribution would yield on average the greatest reduction in the entropy of the target location distribution. In [6],

the sensor node which will result in the smallest expected posterior uncertainty of the target state is chosen as the next node to contribute to the movement decision. Specially, minimizing the expected posterior uncertainty is equivalent to maximizing the mutual information between the sensor node output and the target state [6]. In [7], an entropy-based sensor selection heuristic is proposed for target localization in which a sensor node is chosen at each step and the observation of that node is incorporated into the target location distribution using sequential Bayesian filtering. Instead, the main idea underlying our approach is that the heuristics select an informative sensor such that the fusion of the selected sensor observation with the prior target location distribution would yield to minimize the overall energy consumption in a cluster while maximizing the mutual information between the sensor node observation and the target state in order to improve the quality of the tracking data. We show that properly selected nodes to collect measurements in a cluster head, we can save energy to maximize the sensor network lifetime, and we will compare our node selection algorithm with that of Kaplan [8, 9].

*1.1. Main Contribution.* The main contributions are as follows.

- (i) By using the energy model in [10], we propose a novel energy-based metric to evaluate the energy consumption of node selection algorithms in a cluster of sensor nodes.
- (ii) We formulate the node selection problem as a cross-layer optimization problem and determine the optimal solution by greedy algorithm.
- (iii) We compare the proposed node selection algorithms with the existing literature.
- (iv) We implement distributed tracking algorithms using particle filter method and we compare them with the existing literature.

Preliminary results of this work were published in the author references [11]. The remainder of this paper is organized as follows: in Section 1.2 we describe the existing work. Section 2 provides the preliminary information by describing the model of the overall system, by introducing the energy-based metrics. Section 3 provides a concise review of some basic concepts and statistical model of the nonlinear dynamical system. Section 4 describes the tracking algorithms and formulates the distributed tracking problem introducing a node selection rule. In Section 5 we describe the optimization problem and provide solutions. Section 6 discusses the performances of proposed algorithms, while in Section 7, we draw the main conclusions.

*1.2. Related Work.* Many criteria influence the design of energy-efficient tracking approaches, and a wide range of schemes have been proposed. This Section is devoted to provide an overview of existing energy-efficient techniques for tracking a target in a wireless sensor network.

Generally, the *hierarchical structures* include tree-based, cluster-based, and prediction-based structures. The *tree-based* approaches [12–15] use a hierarchy tree to represent the sensors and record information about presence of the objects being detected by the sensors. Kung and Vlah [12] propose STUN (Scalable Tracking Using Networked Sensors), a scalable tracking architecture that employs hierarchical structure to allow the system to handle a large number of tracked objects. Additionally to the tree, Lin and Tseng [13] consider an in-network moving object tracking in a sensor network, consisting of two operations: location update and query. The drawback is the building of the tree as the target moves. Zhang and Cao [14, 15] propose DCTC (Dynamic Convoy Tree-Based Collaboration). They introduce a message-pruning tree structure called *convoy tree*, which is dynamically configured to add and prune some nodes as the target moves and the tracking problem is formalized as a multiple objective optimization problem. The solution to the problem is a convoy tree sequence with high tree coverage and low energy consumption. Building such a convoy tree sequence requires global network information, and reconfiguration and maintenance of a convoy tree incurs considerable computational and communication overhead. As a result, the tree-based approaches are usually centralized and applied in the deployment phase of sensor networks.

Wang et al. [16] and Chen et al. [17] propose *cluster-based* tracking schemes. They envision a hierarchical sensor network that is composed of (a) a static backbone of sparsely placed position-aware sensors which will assume the role of a cluster head (CH) upon triggered by certain signal events and (b) moderately to densely populated low-end sensors whose function is to provide sensor information to CHs upon request. In these schemes, sensors are grouped into clusters either statically or dynamically (upon detection of the target in the vicinity), and a cluster head collects information from its cluster members and determines the target location using either the trilateration technique [16] or the Voronoi diagram-based approach [17]. Both localization approaches aim to determine the exact location of the target at the expense of considerable computational overhead because of the potentially high number of nodes in the cluster.

From the *topology* perspective, the tracking approaches could use a global or local knowledge about the location of every node in the network. As opposed to the tree-based schemes [12–15] that use a global information, the cluster-based schemes [16, 17] rely on local topology knowledge to limit the scope of target's location updates.

As to the *signal processing*, the tracking approaches can be classified as centralized or distributed. Usually the tree-based schemes are centralized approaches, while the cluster-based schemes are distributed schemes in which the cluster head is the *leader node* in the processing. The works in [13, 18] are centralized approaches, while those in [3, 4, 14, 16, 19] are distributed approaches.

As defined in [20] the *sensor management* is the process of dynamically retasking sensors in response to an evolving environment. The goal of sensor management is to choose actions for individual sensors dynamically so as to maximize overall network utility. In a tracking task the sensor

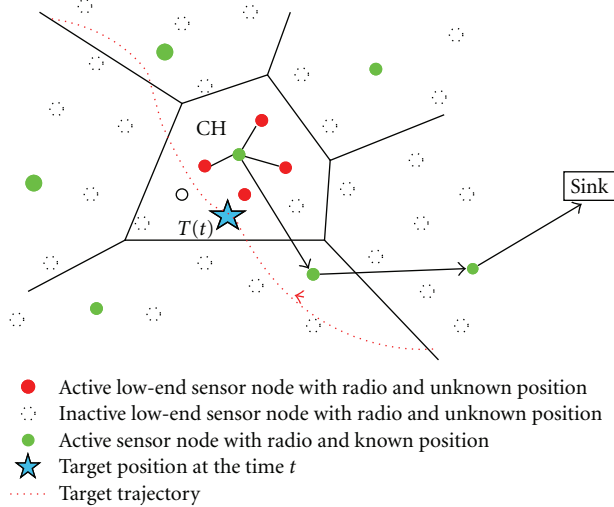


FIGURE 1: Sensor network topology.

management addresses the problem of choosing informative sensors needed to obtain information about the target state and therefore maximize the network lifetime. Based on the *collaboration*, the existing approach of target tracking can be classified in information-driven and information-based. Zhao et al. [19] propose IDSQ (Information Driven Sensor Querying), in which the selection of the best node is based on a Mahalanobis distance that leads to a heuristic method favoring the sensors whose Euclidean distance to the target is small. In [7–9] the node selection problem has been addressed using an information-based approach. The main idea behind this approach is to optimize a utility function, representing the location accuracy, using entropy-based metrics.

As will be discussed in detail in the next sections, the main idea underlined our proposed information-theoretic approach is to optimize an utility function representing the overall energy in a cluster, using energy-based metrics, jointly to the mutual information between the sensor node observation and the target state.

## 2. System Model

We make the following assumptions about the sensor network. First, the network is composed of a single gateway (sink) node and multiple sources. Next, the network is modeled as a combination of (1) a static backbone of sensor nodes aware of their position which assume the role of a cluster head and (2) randomly distributed low-end sensors which sense a moving target and report data to CHs upon request. Finally, we assume that the network is composed of dynamic clusters, depending on the predicted target trajectory (see Figure 1). Indeed, to facilitate collaborative data processing in target tracking sensor networks, usually the sensors are aggregated in clusters, led by a CH.

The details of the clustering algorithm are out of the aim of this paper. In the following we will limit ourselves to consider only the intra-cluster communication issues.

We only highlight that to waste less power the cluster-head should be selected with a shorter distance to the target. The active sensor of the cluster, which will assume the role of cluster head, will predict the trajectory of a target by means of a particle filter based on the history of the target location and some observations which come from some active sensors in the cluster. These sensors in the cluster are chosen to minimize the overall energy consumption.

**2.1. Energy Model.** The main concept underlying the proposed energy efficient tracking algorithm is that the CH should select the active neighbors with the goal of reducing the total energy needed to transmit its data through its radio. In this subsection we introduce two metrics based on energy consumption for a tracking task.

**2.1.1. Energy-Based Metric.** To describe the energy consumption of a tracking algorithm for power-constrained sensor network, we use the energy model for wireless sensor networks introduced by Heinzelman et al. [10]. The energy consumption per bit at the physical layer is

$$E = E_{Tx\text{-elec}} + \beta d^\alpha + E_{Rx\text{-elec}}, \quad (1)$$

where  $E_{Tx\text{-elec}}$  is a distance independent term that takes into account overheads of transmitter electronics (PLLs, VCOs, bias currents, etc.) and digital processing;  $E_{Rx\text{-elec}}$  is a distance independent term that takes into account overheads of receiver electronics; finally  $\beta d^\alpha$  accounts for the radiated power necessary to transmit one bit over a distance  $d$  between source and destination, where  $\alpha$  is the exponent of the path loss ( $2 \leq \alpha \leq 5$ ).

According to [21] we assume that  $E_{Tx\text{-elec}} = E_{Rx\text{-elec}} = E_{elec}$ . Hence, given  $l$  bits of data, the overall energy consumption to transmit the packet of  $l$  bit between two nodes at a distance  $d$  with a given received SNR can be expressed as

$$E(d, l) = (2E_{elec} + E_{amp} \cdot d^\alpha) \cdot l, \quad (2)$$

where  $E_{elec}$  [Joule/bit] is the energy needed by the transceiver circuitry to transmit or receive one bit and  $E_{amp}$  [Joule/(bit · m $^\alpha$ )] is a constant which represents the energy needed to transmit one bit over a distance  $d$  to achieve an acceptable SNR at the destination.

This model assumes that the energy consumption is dominated by the radio communication rather than the computation. We refer to (2) as the energy-based metric.

**2.1.2. Residual Energy-Based Metric.** According to [22, 23] we propose another metric combining energy and remaining energy at nodes. Hence, if we refer to a link  $(i, j)$  with distance  $d$ , the overall energy consumption to transmit a  $l$ -bit packet between the node  $i$  and node  $j$  at a distance  $d$  with a given received SNR can be expressed as

$$E_{ij}(d, l) = E_r - (2E_{elec} + E_{amp} \cdot d^\alpha) \cdot l, \quad (3)$$

where the first addend  $E_r$  is the remaining energy at node  $i$  and the second addend is the transmission energy required for the node  $i$  to transmit  $l$  bits to its neighboring node  $j$ . We refer to (3) as the residual energy-based metric.

**2.2. Network Model.** In our analysis, we consider a network composed of randomly deployed sensor nodes which sense a moving target and forward the information into a position-aware sensor which acts as a data gathering node. The network is divided into  $N_c$  clusters each having  $N_a$  nodes. Each sensor is equipped by a low data rate radio interface. The position-aware sensors are equipped by two radio transmitters, that is, a low data rate transmitter to communicate with the sensors, and a high rate wireless interface for CH-CH communication.

In our analysis, we assume to know the position of the CH (static node) and to estimate the distance of each neighbor with respect the CH. Many types of sensors provide measurements that are function of the relative distance between the sensor and the sensed object (e.g., acoustic sensors, sonar, etc.). We consider a common example, newly in [4], where sensors measure the power of a radio signal emitted by the object.

Therefore, we assume the log-normal shadowing model for the channel and we suppose that the power of a received signal decreases exponentially with the propagation distance:

$$P_r(d) = P_r(d_o) \cdot \left(\frac{d_o}{d}\right)^\alpha + X_\sigma, \quad (4)$$

where  $P_r(d)$  is the received power at a receiver at distance  $d$  from a transmitter,  $P_r(d_o)$  is the transmitted power at a reference distance  $d_o$ ,  $\alpha$  is the path loss exponent ( $2 \leq \alpha \leq 5$ ), and  $X_\sigma$  is the shadow fading component, with  $X_\sigma$  Gaussian distribution  $\mathcal{N}(0, \sigma)$ .

Hence, the distance from the  $i$ th sensor of the cluster to the CH can be estimated as  $d = (P_r/P_a)^{-1/\alpha}$ , where  $P_r$  is the received signal strength in the sensor, and  $P_a$  is the (unknown) strength of the signal from the sensor.

### 3. Basic Concepts and Theoretical Bound

We begin our analysis with a concise review of some basic concepts and the models of a nonlinear dynamical system.

We consider the state estimation of a nonlinear dynamical system:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{f}_k(\mathbf{x}_k, \mathbf{w}_k), \\ \mathbf{z}_k &= \mathbf{h}_k(\mathbf{x}_k, \mathbf{v}_k), \end{aligned} \quad (5)$$

where,  $k$  is the discrete time index;  $\mathbf{x}_k \in \mathbf{R}^n$  is the state vector;  $\mathbf{z}_k \in \mathbf{R}^m$  is the observation vector;  $\mathbf{w}_k$  is the zero-mean white Gaussian process noise with nonsingular covariance matrix  $\mathbf{Q}_k$ ;  $\mathbf{v}_k$  is the zero-mean white Gaussian measurement noise independent of  $\mathbf{w}_k$  with nonsingular covariance matrix  $\mathbf{R}_k$ ;  $\mathbf{f}_k$  and  $\mathbf{h}_k$  are the (in general) nonlinear functions.

The functions  $\mathbf{f}_k$  and  $\mathbf{h}_k$  may depend on time  $k$ . Further we assume that the initial state  $\mathbf{x}_0$  has a known probability density function  $p(\mathbf{x}_0)$ . Let  $\nabla$  and  $\Delta$  be operators of the first- and second-order partial derivatives, respectively,

$$\begin{aligned} \nabla_{\mathbf{x}_k} &= \left[ \frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_r} \right]^T, \\ \Delta_{\mathbf{x}_k}^{\mathbf{x}_k} &= \nabla_{\mathbf{x}_k} \nabla_{\mathbf{x}_k}^T. \end{aligned} \quad (6)$$

Using this notation, the Fisher Information Matrix (FIM) given by [24] for this problem can be written as

$$\mathbf{J}(\mathbf{x}_k) = -E \left[ \Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \log p(\mathbf{z}_k, \mathbf{x}_k) \right], \quad (7)$$

where  $p(\mathbf{z}_k, \mathbf{x}_k)$  represents the joint probability density of  $(\mathbf{z}_k, \mathbf{x}_k)$ . In the following sections  $\mathbf{J}(\mathbf{x}_k)$  is denoted by  $\mathbf{J}_k$  for brevity. The following proposition [25] gives an efficient method for computing  $\mathbf{J}_k$  recursively.

**Proposition 1.** *The sequence  $\{\mathbf{J}_k\}$  of Fisher information submatrices for estimating state vectors  $\{\mathbf{x}_k\}$  obeys the recursion:*

$$\mathbf{J}_{k+1} = \mathbf{D}_k^{22} - \mathbf{D}_k^{21} \left( \mathbf{J}_k + \mathbf{D}_k^{11} \right)^{-1} \mathbf{D}_k^{12}, \quad (8)$$

where

$$\begin{aligned} \mathbf{D}_k^{11} &= E \left\{ -\nabla_{\mathbf{x}_k} \nabla_{\mathbf{x}_k}^T \log p(\mathbf{x}_{k+1} | \mathbf{x}_k) \right\}, \\ \mathbf{D}_k^{12} &= E \left\{ -\nabla_{\mathbf{x}_{k+1}} \nabla_{\mathbf{x}_k}^T \log p(\mathbf{x}_{k+1} | \mathbf{x}_k) \right\}, \\ \mathbf{D}_k^{21} &= E \left\{ -\nabla_{\mathbf{x}_k} \nabla_{\mathbf{x}_{k+1}}^T \log p(\mathbf{x}_{k+1} | \mathbf{x}_k) \right\} = \left[ \mathbf{D}_k^{12} \right]^T, \\ \mathbf{D}_k^{22} &= E \left\{ -\nabla_{\mathbf{x}_{k+1}} \nabla_{\mathbf{x}_{k+1}}^T \log p(\mathbf{x}_{k+1} | \mathbf{x}_k) \right\} \\ &\quad + E \left\{ -\nabla_{\mathbf{x}_{k+1}} \nabla_{\mathbf{x}_{k+1}}^T \log p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}) \right\}. \end{aligned} \quad (9)$$

**3.1. Time-Invariant Model.** We assume that the functions  $\mathbf{f}_k(\cdot, \cdot)$  and  $\mathbf{h}_k(\cdot, \cdot)$  are time invariant (independent of  $k$ ). Using this assumption and according to [25], the matrices  $\mathbf{D}_k^{11}, \dots, \mathbf{D}_k^{22}$  do not depend on  $k$ , and additionally for  $k \rightarrow \infty$ , the matrix  $\mathbf{J}_k$  converges to a matrix  $\mathbf{J}_\infty$ , which is given as a solution to the following equation:

$$\mathbf{J}_\infty = \mathbf{D}_k^{22} - \mathbf{D}_k^{21} \left( \mathbf{J}_\infty + \mathbf{D}_k^{11} \right)^{-1} \mathbf{D}_k^{12}. \quad (10)$$

Note that (10) is a discrete-time algebraic Riccati equation. A common form of the Riccati equation is obtained if the recursion (8) is equivalently written as

$$\begin{aligned} \mathbf{J}_{k+1} &= \mathbf{D}_k^{21} \left( \mathbf{D}_k^{11} \right)^{-1} \mathbf{J}_k \left( \mathbf{D}_k^{11} \right)^{-1} \mathbf{D}_k^{12} - \mathbf{D}_k^{21} \left( \mathbf{D}_k^{11} \right)^{-1} \\ &\quad \cdot \mathbf{J}_k \left( \mathbf{J}_k + \mathbf{D}_k^{11} \right)^{-1} \mathbf{J}_k \left( \mathbf{D}_k^{11} \right)^{-1} \mathbf{D}_k^{12} \\ &\quad + \mathbf{D}_k^{22} - \mathbf{D}_k^{21} \left( \mathbf{D}_k^{11} \right)^{-1} \mathbf{D}_k^{12}, \end{aligned} \quad (11)$$

which can be easily proved by simple algebraic manipulations. Then, put  $\mathbf{J}_{k+1} = \mathbf{J}_k = \mathbf{J}_\infty$ .

**3.2. Model for State Estimation of a Nonlinear Dynamical System.** The problem of estimating the state vector of a nonlinear dynamical system (i.e., single target tracking) can be formulated as follows. The state and the observations of the target of interest are assumed to follow the following model:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{F}_k \mathbf{x}_k + \mathbf{A}_k \mathbf{w}_k, \\ \mathbf{z}_k &= \mathbf{H}_k(\mathbf{x}_k) + \mathbf{B}_k \mathbf{v}_k, \end{aligned} \quad (12)$$



where  $\mathbf{x}_k$ ,  $\mathbf{z}_k$ ,  $\mathbf{w}_k$ , and  $\mathbf{v}_k$  were introduced in Section 3. The matrices  $\mathbf{F}_k$ ,  $\mathbf{A}_k$ , and  $\mathbf{B}_k$  are independent of the state vector whereas  $\mathbf{H}_k$  is a function dependent of the state vector  $\mathbf{x}_k$ . From the assumption that the noises  $\mathbf{w}_k$  and  $\mathbf{v}_k$  are Gaussian with zero mean and invertible covariances matrices  $\mathbf{Q}_k$  and  $\mathbf{R}_k$ , respectively, and that the elements of the noise vectors  $\mathbf{w}_k$  and  $\mathbf{v}_k$  are independent and identically distributed (IID) it follows that the conditional densities can be written as

$$\begin{aligned} p(\mathbf{x}_{k+1} | \mathbf{x}_k) &= \frac{1}{\sqrt{2\pi}|\mathbf{Q}_i|} \cdot e^{-(1/2)[\mathbf{x}_{k+1}-\mathbf{F}_k\mathbf{x}_k]^T\mathbf{Q}_k^{-1}[\mathbf{x}_{k+1}-\mathbf{F}_k\mathbf{x}_k]}, \\ p(\mathbf{z}_k | \mathbf{x}_k) &= \frac{1}{\sqrt{2\pi}|\mathbf{R}_i|} \cdot e^{-(1/2)[\mathbf{z}_k-\mathbf{H}_k(\mathbf{x}_k)]^T\mathbf{R}_k^{-1}[\mathbf{z}_k-\mathbf{H}_k(\mathbf{x}_k)]} \end{aligned} \quad (13)$$

In the derivation above, we assume that the system model is time-invariant, implying that  $\mathbf{Q}_1 = \mathbf{Q}_2 = \dots = \mathbf{Q}_k = \mathbf{Q}$  and  $\mathbf{R}_1 = \mathbf{R}_2 = \dots = \mathbf{R}_k = \mathbf{R}$ . Consequently, from (13), we have

$$\begin{aligned} -\log p(\mathbf{x}_{k+1} | \mathbf{x}_k) &= c_1 + \frac{1}{2}[\mathbf{x}_{k+1} - \mathbf{F}_k\mathbf{x}_k]^T\mathbf{Q}_k^{-1}[\mathbf{x}_{k+1} - \mathbf{F}_k\mathbf{x}_k], \\ -\log p(\mathbf{z}_k | \mathbf{x}_k) &= c_2 + \frac{1}{2}[\mathbf{z}_k - \mathbf{H}_k(\mathbf{x}_k)]^T\mathbf{R}_k^{-1}[\mathbf{z}_k - \mathbf{H}_k(\mathbf{x}_k)], \end{aligned} \quad (14)$$

where  $c_1$  and  $c_2$  are constants, and

$$\begin{aligned} \mathbf{D}_k^{11} &= E\left\{\nabla_{\mathbf{x}_k}[\mathbf{x}_{k+1} - \mathbf{F}_k\mathbf{x}_k]^T\mathbf{Q}_k^{-1} \cdot \nabla_{\mathbf{x}_k}^T[\mathbf{x}_{k+1} - \mathbf{F}_k\mathbf{x}_k]\right\}, \\ \mathbf{D}_k^{12} &= -E\left\{\nabla_{\mathbf{x}_k}[\mathbf{x}_{k+1} - \mathbf{F}_k\mathbf{x}_k]^T\right\}, \\ \mathbf{D}_k^{22} &= \mathbf{Q}_k^{-1} + E\left\{-\nabla_{\mathbf{x}_{k+1}}[\mathbf{z}_{k+1} - \mathbf{H}_{k+1}(\mathbf{x}_{k+1})]^T \cdot \mathbf{R}_{k+1}^{-1}\nabla_{\mathbf{x}_{k+1}}^T[\mathbf{z}_{k+1} - \mathbf{H}_{k+1}(\mathbf{x}_{k+1})]\right\}. \end{aligned} \quad (15)$$

#### 4. Tracking Algorithms

The aim of this section is to describe the proposed cross-layer predictive tracking algorithm and to discuss that it can consistently reduce the energy consumption on sensors with respect to existent one-level location algorithms. As stated above, in this paper we use sequential Monte Carlo (SMC) approaches, also known as particle filtering [26, 27], for tracking a moving target while Kaplan [8, 9] estimates the target location using a Kalman filter based on the current measurement at a sensor and the past history at other sensors. The particle filter provides simulation-based solutions to estimate the posterior distribution of nonlinear discrete time dynamic models. The main idea of particle filtering is to represent the required posterior distribution density by a set of random samples with associated weights and to compute estimates based on these samples and weights, updating them recursively in time using the sequential importance sampling (SIS) algorithm. As the number of samples becomes very large, the SIS filter approaches the optimal Bayesian estimate. A common problem with the SIS particle filter is the degeneracy phenomenon, since after a few iterations all the particles, with exception of one of them, will have negligible weight. Because of this phenomenon, resampling techniques

are used to eliminate particles that have small weights and to concentrate on particles with large weights. The particle filter using sequential importance resampling (SIR) techniques is known as *bootstrap filter* or SIR particle filter (PF). Therefore, in the following we use the SIR filter to achieve the tracking task. Moreover, we compare the performance of bootstrap filter with the unscented particle filter. Indeed, unfortunately even when resampling schemes are used, degeneracy may still be a problem. Using the prior distribution as importance distribution could lead to the degeneracy problem of the particles because the most recent observations are ignored. Samples may eventually collapse to a single point if, during the resampling stage, samples with high importance weights are duplicated an extremely large number of times. There have been numerous proposals to rectify the degeneracy problem improving the performance of the SIR particle filter [28]. Notable techniques include local linearization using the extended Kalman filter (EKF) [29, 30] or the unscented Kalman filter (UKF) to estimate the importance distribution [31]. A particle filter which uses UKF to generate the importance distribution is referred as *unscented particle filter* (UPF) or sigma-point particle filter [31].

In this paper, we analyze the energy efficiency of particle filtering looking at collaborative and distributed schema for tracking a moving target.

**4.1. Node Selection.** We formulate the problem of distributed tracking as a sequential Monte Carlo estimation problem. Assume that the state of a target we wish to estimate is  $\mathbf{x}_k$ . Each new sensor measurement  $\mathbf{z}_k$  is combined with the current estimate  $p(\mathbf{x}_k | \mathbf{z}_1, \dots, \mathbf{z}_{k-1})$ , hereafter called belief state, to form a new belief state of the target  $p(\mathbf{x}_k | \mathbf{z}_1, \dots, \mathbf{z}_k)$ . The problem of selecting a sensor in order to provide greatest improvement to the estimation at the lowest cost becomes an optimization problem. We let  $Z_k$  be all measurements that have already been used at time  $k$  in the inference of the current belief state and refer the sensor which holds the belief state as the leader node. The objective function for this optimization problem can be defined as a mixture of both information gain and cost. In the remainder of the section we consider the information gain, while the computation of the energy consumption cost is discussed in the next section.

The information gain to select the sensor  $s$  can be defined as  $\tilde{\Phi}_s(p(\mathbf{x}_k | Z_k)) = \Phi_{\text{Utility}}(p(\mathbf{x}_{k+1} | Z_k, \mathbf{z}_{k+1,s}))$ , where  $\mathbf{z}_{k+1,s}$  is the new measurement from sensor  $s$  at time  $k+1$ . The utility function can be defined as the uncertainty of the target state reduced by the additional measurements  $\mathbf{z}_{k+1,s}$  [6], that is,  $\tilde{\Phi}_s(p(\mathbf{x}_k | Z_k)) = H_{\text{target}}(Z_k) - H_{\text{target}}(Z_k, \mathbf{z}_{k+1,s})$ . Furthermore, the utility function can be defined with the mutual information  $\Phi_{\text{Utility}}(p(\mathbf{x}_{k+1}, \mathbf{z}_{k+1,s} | Z_k)) = I(\mathbf{x}_{k+1}, \mathbf{z}_{k+1,s} | Z_k)$ , which means the information of  $\mathbf{x}_{k+1}$  conveyed by the new measurements  $\mathbf{z}_{k+1,s}$  [18].

The utility function based on the entropy is difficult to compute in practice since we need to have the measurement before deciding how useful it is. Instead of the true *a posteriori* distribution, a more practical alternative is to compute the entropy based on the expected posterior distribution. In the ideal case when a real new measurement  $\mathbf{z}_{k+1,s}$  is

available, the new belief or posterior is evaluated using sequential Bayesian filtering  $p(\mathbf{x}_{k+1} | \mathbf{z}_{k+1,s}, Z_k) \propto p(\mathbf{z}_{k+1,s} | \mathbf{x}_{k+1})p(\mathbf{x}_{k+1} | Z_k)$ . Without having the data  $\mathbf{z}_{k+1,s}$ , we need to compute the expected posterior distribution  $E_{\mathbf{z}_{k+1,s}}(p(\mathbf{x}_{k+1} | \mathbf{z}_{k+1,s}, Z_k))$ . We can estimate the measurement  $\bar{\mathbf{z}}_{k+1,s}$  from the predicted belief and compute the expected likelihood function  $\hat{p}(\bar{\mathbf{z}}_{k+1,s} | \mathbf{x}_{k+1}) = \int p(\mathbf{z}_{k+1,s} | \mathbf{x}_{k+1}) \times p(\mathbf{v}_{k+1} | Z_k) d\mathbf{v}_{k+1}$ . Then, the expected posterior belief can be defined as follows:

$$\hat{p}(\mathbf{x}_{k+1} | \bar{\mathbf{z}}_{k+1,s}, Z_k) = \hat{p}(\bar{\mathbf{z}}_{k+1,s} | \mathbf{x}_{k+1})p(\mathbf{x}_{k+1} | Z_k). \quad (16)$$

The entropy of expected posterior distribution can be computed based on the discrete belief state  $\{\mathbf{x}_k^j, w_k^j\}_{j=1}^N$  [32], where  $w_k^j$  is the importance sampling weight in the resampling step of the particle filters and  $N$  represents the number of weights namely the number of particles. According to (16), the expected posterior belief for sensor  $s$  can be represented by the discrete belief state  $\{\mathbf{x}_k^j, \tilde{w}_{k+1,s}^j\}_{j=1}^N$  with the weights  $\tilde{w}_{k+1,s}^j$  given by

$$\tilde{w}_{k+1,s}^j = \hat{p}(\bar{\mathbf{z}}_{k+1,s} | \mathbf{x}_{k+1}^j) w_{k+1}^j. \quad (17)$$

The entropy of the discrete belief state  $\{\mathbf{x}_{k+1}^j, \tilde{w}_{k+1,s}^j\}_{j=1}^N$  can be computed by

$$H = - \sum_{j=1}^N \tilde{w}_{k+1,s}^j \log \tilde{w}_{k+1,s}^j. \quad (18)$$

This expected posterior entropy can be used as a criteria to select the best among the sensor candidates to maximize the information gain. The objective function expected to improve the estimation of the target is given by

$$\begin{aligned} \mathcal{N}_s &= \arg \max_{s \in \mathcal{N}_a} \tilde{\Phi}_s(p(\mathbf{x}_k | Z_k)) \\ &= \arg \max_{s \in \mathcal{N}_a} (H_{\text{target}}(Z_k) - H_{\text{target}}(Z_k, \mathbf{z}_{k+1,s})), \end{aligned} \quad (19)$$

where  $\mathcal{N}_a$  indicates the set, with cardinality  $N_a$ , of active nodes in the cluster that receive from the CH a signal exceeding a predetermined RSS threshold.

## 5. Energy Efficient Tracking

Let us consider the following location discovery protocol for a given snapshot. With reference to Figure 2, the target  $T$  periodically sends discovery signal, with period  $T_M$ , to all the sensors of the network. We indicate with  $\mathcal{N}_s$  the set of active neighbor nodes that maximize the utility function as in (19). The subset  $\mathcal{N}_d$  of desired anchor nodes needed for the localization algorithm will be chosen so as to minimize the energy consumption of the location discovery protocol. Then, each node  $i \in \mathcal{N}_s$  transmits the sensing information (the distance of the node  $i$  from the target) to the CH which processes the data and updates the current target location.

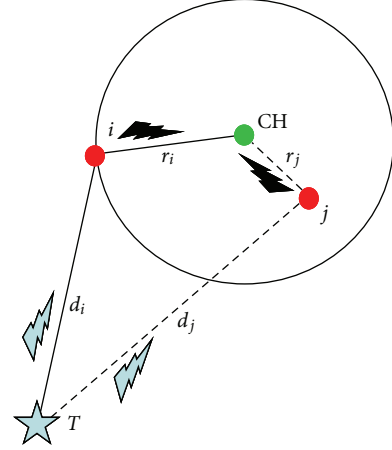


FIGURE 2: Location discovery protocol.

According to metric (2), the energy cost in the communication with one sensor  $i \in \mathcal{N}_s$  is given by

$$\begin{aligned} E_i(r_i, d_i) &= [E_{\text{elec}}(N_d + 2) + E_{\text{amp}} \cdot d_i^\alpha] \cdot \frac{l}{T_M} \\ &+ [E_{\text{elec}}(N_d + 1) + E_{\text{amp}} \cdot r_i^\alpha] \cdot b, \end{aligned} \quad (20)$$

where  $b$  represents the bit rate (bit/s) between the CH and the neighbor  $i$ ,  $T_M$  (s) is the period between two consecutive discovery signal of the target,  $r_i$  and  $d_i$  are, respectively, the distance of the node  $i$  from the CH and the target, and  $N_d$  is the number of desired neighbors of the CH. Finally, in (20)  $E_{\text{elec}}N_d$  represents the energy needed at the neighbors to receive one bit. In the energy cost we have omitted the energy consumption in the path between the target and the CH due to the calibration phase of the clustering and we have considered only the communication between each node of the cluster and its cluster head.

On the other hand, according to the metric (3) and the location discovery protocol herein discussed, the energy cost in the communication with the sensor  $i \in \mathcal{N}_s$ , based on (20), is given by

$$U_i(r_i, d_i) = 2E_r(i) - E_i(r_i, d_i). \quad (21)$$

The total utility function, for all nodes in the set  $\mathcal{N}_s$ , is given by

$$U^{\text{TOT}}(\mathcal{N}_s) = \sum_{i \in \mathcal{N}_s} U_i(r_i, d_i). \quad (22)$$

As stated above, our objective is to select the optimal subset  $\mathcal{N}_d \subset \mathcal{N}_s$  which maximizes the total utility function in (22), subject to a constraint of the cardinality  $N_d$  of said subset. This gives formally the following objective function and associated constraint:

$$\mathcal{N}_d = \arg \max_{\mathcal{N} \subseteq \mathcal{N}_s} U^{\text{TOT}}(\mathcal{N}) \quad \text{subject to } N_d \geq 2 \quad (23)$$

A unique solution to this problem exists, since the objective function is strictly concave and the feasible set is convex. The solution of this optimization problem is illustrated in the following subsections.

**Synopsis:**  $[\mathcal{N}_d, \mathcal{C}, E_{\text{tot}}, \text{node}, E_b, k] = \text{Greedy}(\mathcal{N}_s, \mathcal{N}_d)$ .  
**Given:** Set of nodes in the cluster  $\mathcal{N}_s$ , number of desired nodes  $\mathcal{N}_d$ .  
**Output:** Set of desired nodes  $\mathcal{N}_d$ , new set of candidate nodes in the cluster  $\mathcal{C}$ , total energy of the desired set  $E_{\text{tot}}$ , last node selected in the current snapshot  $\text{node}$ , energy of this node  $E_b$ , time  $k$ .

```

Initialize the candidate set:
     $\mathcal{C} = \mathcal{N}_s$ 
     $\mathcal{N}_{\text{cand}} = 0$ 
Initialize the objective function:
     $E_{\text{tot}} = 0$ 
Randomly select a candidate node  $i \in \mathcal{C}$ 
     $E_{\text{min}} = E(i)$ 
     $\text{NodeMin} = i$ 
while  $|\mathcal{N}_{\text{cand}}| < \mathcal{N}_d$  do
    for each  $j \in \mathcal{C} \setminus \{i\}$  do
        if  $E(j) < E_{\text{min}}$  then
             $E_{\text{min}} = E(j)$ 
             $\text{NodeMin} = j$ 
        end if
    end for
     $\text{node} = \text{NodeMin}$ 
     $E_b = E_{\text{min}}$ 
     $E_{\text{tot}} = E_{\text{tot}} + E_{\text{min}}$ 
     $\mathcal{N}_{\text{cand}} = \mathcal{N}_{\text{cand}} \cup \{\text{NodeMin}\}$ 
     $\mathcal{C} = \mathcal{C} \setminus \{\text{NodeMin}\}$ 
end while
 $\mathcal{N}_d = \mathcal{N}_{\text{cand}}$ 
    
```

ALGORITHM 1: Greedy random node selection.

5.1. *The Solution in the Static Scenario.* To find the optimal solution for such problem it is theoretically possible to enumerate the solutions and evaluate each with respect to the stated objective. However, from a practical perspective, it is infeasible to follow such a strategy because the number of combinations grows exponentially with the size of problem.

Indeed, if we formulate our combinatorial optimization problem as an integer linear programming problem, the computational complexity consists of enumerating all the  $N_d$ -node subsets,  $O(N_a^{N_d})$ , and adding the computational complexity of the assignment problem,  $O(N_d^3)$ . In such cases, heuristic methods are usually employed to find good, but not necessarily guaranteed optimal solutions. More than one technique is applicable, that is, integer linear programming, graph theory, genetic algorithms, and greedy heuristics; see [33] for further details. Here we adopt the meta-heuristic greedy randomized adaptive search procedures (GRASP) [34], in which each iteration consists of two phases, a construction phase, in which a feasible solution is produced, and a local search phase, in which a local optimum in the neighborhood of the constructed solution is sought. The best overall solution is kept as the result.

The implementation of the optimal Greedy node selection procedure is described in Algorithm 1. It provides the set

**Synopsis:**  $[\mathcal{N}_d, \mathcal{C}', E_{\text{tot}}] = \text{BranchBound}(N_d, \mathcal{C}, E_b, \text{node})$ .  
**Given:** Number of desired nodes  $N_d$ , candidate nodes set of the cluster  $\mathcal{N}_a$ , energy bound  $E_b$ , node related to the energy bound  $\text{node}$ .  
**Output:** Set of desired nodes  $\mathcal{N}_d$ , new set of candidate nodes in the cluster  $\mathcal{C}'$ , total energy of the desired set  $E_{\text{tot}}$ .

```

Initialize the candidate set:
     $\mathcal{N}_{\text{cand}} = \{\text{node}\}$ 
Initialize the objective function:
     $E_{\text{tot}} = E_b$ 
while  $|\mathcal{N}_{\text{cand}}| < N_d$  do
    for each  $j \in \mathcal{C} \setminus \{i\}$  do
        if  $E(j) < E_b$  then
            break
        else
             $E_{\text{min}} = E(j)$ 
             $\text{NodeMin} = j$ 
        end if
    end for
     $\mathcal{N}_{\text{cand}} = \mathcal{N}_{\text{cand}} \cup \{\text{NodeMin}\}$ 
     $\mathcal{C} = \mathcal{C} \setminus \{\text{NodeMin}\}$ 
     $E_{\text{tot}} = E_{\text{tot}} + E_{\text{min}}$ 
end while
 $\mathcal{N}_d = \mathcal{N}_{\text{cand}}$ 
    
```

ALGORITHM 2: Branch and bound algorithm.

of desired nodes  $\mathcal{N}_d$ , new set of candidate nodes in the cluster  $\mathcal{C}$ , total energy of the desired set  $E_{\text{tot}}$ , last node selected in the current snapshot  $\text{node}$ , energy of this node  $E_b$ , and time  $k$ .

5.2. *The Solution in the Dynamic Scenario.* In previous sections, we have considered the static version of the problem, namely, a snapshot model. In this section we extend the Greedy node selection procedure over multiple snapshots, so that we can select active nodes for the next measurement intervals. In a dynamic scenario, due to the target mobility, the distance  $d_i$  in (20) varies with the time and hence the total utility function (22) is a function of time  $k$ .

In the dynamic version of the optimization problem we use the Dynamic Programming [35], that is based on the idea of breaking down the problem into stages at which the decisions take place and finding a recurrence relation that takes us backward from one stage to the previous stage. For this purpose, a branch-and-bound method is developed, in which the branch refers to the partitioning process into stages, that are repeatedly decomposed until a solution is found or infeasibility is proved, and the bound refers to lower bounds that are used to construct a proof of optimality without exhaustive search. We introduce an *energy bound* as in the following definition.

*Definition 1.* The energy bound is the maximum energy referred to the energy costs associated to the nodes selected in the previous their snapshot.

**Synopsis:**  $[\mathcal{N}_d, \mathcal{C}, E_{\text{tot}}] = \text{DynamicSelection}(\mathcal{N}_a, N_d)$ .  
**Output:** Set of desired nodes  $\mathcal{N}_d$ , new set of candidate nodes in the cluster  $\mathcal{C}$ , total energy of the desired set  $E_{\text{tot}}$ .

- (1) The initial leader node does the following step:
  - (a) draw initial samples  $\{\mathbf{x}_0^j, w_0^j = 1\}_{j=1}^N$  of the target from the prior information; (b) update the belief state  $\{\mathbf{x}_1^j, w_1^j\}_{j=1}^N$  by the sensor fusion algorithm based on the new measurement  $\mathbf{z}_1$  at the leader node in the set  $\mathcal{N}_a$ ; (c) compute the expected posterior belief state  $\{\mathbf{x}_2^j, w_{2,i}^j\}_{j=1}^N$  for each neighbor node  $i$  with the weights  $\tilde{w}_{2,i}^j$  computed by (17); (d) compute the entropy of the expected posterior belief state  $\{\mathbf{x}_2^j, w_{2,i}^j\}_{j=1}^N$  for each neighbor node  $i$  by (18) and determine the next best sensor, say (b) in the set  $\mathcal{N}_s$ .
- (2)  $[\mathcal{N}_d, \mathcal{C}, E_{\text{tot}}, \text{node}, E_b, k] = \text{Greedy}(\mathcal{N}_s, N_d)$
- (3) Loop until time runs out:
- (4) Prediction step and Update step of particle filtering to estimate the target's trajectory.
- (5) Update the candidate set  $\mathcal{C}$  during the dynamic of the target.
- (6)  $[\mathcal{N}_d, \mathcal{C}', E_{\text{tot}}] = \text{BranchBound}(N_d, \mathcal{C}, E_b, \text{node})$

ALGORITHM 3: Tracking algorithm.

TABLE 1: Computational complexity of the node selection algorithms.

Number of desired nodes	Greedy	Kaplan
1	$O(N_a - 1)$	$O((N_a - 1)^2)$
2	$O(N_a - 2)$	$O((N_a - 2)^2)$
...	...	...
$N_d$	$O(N_a - N_d)$	$O((N_a - N_d)^2)$

Algorithm 2 shows pseudocode of an efficient implementation of our branch-and-bound approach. It provides the set of desired nodes  $\mathcal{N}_d$ , the new set of candidate nodes in the cluster  $\mathcal{C}'$ , and the total energy of the desired set  $E_{\text{tot}}$ . Note that the total utility function of the desired set can be obtained from the following equation:

$$U^{\text{TOT}}(\mathcal{N}_d) = 2E_r N_d - E^{\text{TOT}}(\mathcal{N}_d), \quad (24)$$

assuming that the nodes of the cluster have an even remaining energy, that is,

$$E_r(i) = E_r \quad \forall i \in \mathcal{N}_d. \quad (25)$$

Finally, in Algorithm 3 has been reported the overall tracking algorithm which combines the node's selection procedures with the particle filtering algorithm.

## 6. Performance Evaluation

In this section, we investigate the performance of the overall target tracking system looking first at the node selection

TABLE 2: Time to process greedy and Kaplan algorithms.

Number of desired nodes	Greedy Time	Kaplan Time
2	$6.6639e - 5$	$13.9816e - 4$
3	$7.8829e - 5$	$32.5238e - 4$
4	$1.0095e - 4$	$10.0543e - 3$
5	$1.2099e - 4$	$23.1352e - 3$
6	$1.4041e - 4$	$52.2764e - 3$

algorithm and then at the tracking algorithm and energy consumption.

**6.1. Optimal Node Selection.** In the following we compare our *greedy* node selection algorithm with the *Kaplan* algorithm in [8, 9]. The only difference between [8] and [9] is that in [8] the global topology knowledge is assumed, in which every active node reaches the entire network, while in [9] the only knowledge of the relative position to the target and the active nodes from the previous snapshot is required. Table 1 shows the computational complexity of the proposed node selection algorithm and the *Kaplan* algorithm for each iteration. Hence, the computational complexity for all iterations is given by the following.

- (i) Greedy computational complexity:  $\sum_{i=0}^{N_d} (N_a - i) = N_a N_d + N_a - (N_d^2/2) - (N_d/2)$ .
- (ii) Kaplan computational complexity:  $\sum_{i=0}^{N_d} (N_a - i)^2 = N_a^2(N_d + 1) + (N_d(N_d + 1)(2N_d + 1))/(6) - 2N_a(N_d(N_d + 1))/2$ .

Indeed, in the above analysis we have omitted the computational complexity of the initialization step of the Simplex algorithm, in which two nodes are chosen by exhaustive search. Definitely, due to computational complexity and because the simplex does not always find the global minimum, our approach outperforms the *Kaplan* algorithm.

As stated in Section 4.1, the node selection procedure is combined with the maximization of the utility function. Hence in our analysis the computational complexity of the problem in (19) needs to be considered, namely,  $O(N_d N \log N)$  where  $N$  is the number of weights. Finally, the computational complexity for all iterations is given by  $\sum_{i=1}^{N_d} i N \log N$ .

In Table 2 the results of a runtime measurement are illustrated, conducted on a system with AMD Opteron XP Processor 250, approximately 2400 MHz frequency, and 4,00 GB RAM. Table 2 provides the execution time of the node's selection algorithm versus the number of desired nodes using  $N_a$  equal to 10 and  $N$  equal to 100.

**6.2. Estimation Bound for Range-Based Tracking.** In this paper we assume as dynamic model of the target the constant velocity model [36]. Hence, denoting by  $\mathbf{x}_k = [\alpha_k, \dot{\alpha}_k, \beta_k, \dot{\beta}_k]^T$  the state vector (coordinates along  $x, y$  axes and the velocities) of a target, the state-space model is



TABLE 3: Parameters of the model used for simulations.

Parameters	Scenario 1	Scenario 2
Size	20 m <sup>2</sup>	200 m <sup>2</sup>
velocity	0.1 m/s	0.3 m/s
$K$	9 dB	9 dB
$\alpha$	3	3
$E_{\text{elec}}$	10 nJ/bit	10 nJ/bit
$E_{\text{amp}}$	100 pJ/bit/m <sup>3</sup>	100 pJ/bit/m <sup>3</sup>
$T_M$	2 sec	2 sec
$b$	10 bit/sec	10 bit/sec
$l$	8 bits	8 bits
$\Delta T$	1 sec	1 sec
Process variance	1.0	1.8
Observation variance	0.3	0.3
particle number	100, 500	100, 500

given by

$$\mathbf{x}_{k+1} = \begin{pmatrix} 1 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{x}_k + \begin{pmatrix} \frac{\Delta T^2}{2} & 0 \\ \Delta T & 0 \\ 0 & \frac{\Delta T^2}{2} \\ 0 & \Delta T \end{pmatrix} \mathbf{w}_k, \quad (26)$$

where  $\mathbf{v}_k \sim \mathcal{N}(0, \text{diag}(\sigma_x^2, \sigma_y^2))$  denotes the motion noise and  $\Delta T$  the length of the measurement interval.

Additionally, as observation model of the measurements, we use the log-normal shadowing model [37]. Hence, let  $\{\alpha^s, \beta^s\}$  be the fixed position of sensor  $\mathbf{s}$  and let  $\mathbf{d}_k = \|\mathbf{x}_k - \mathbf{s}\|^{1/2} = [(\alpha_k - \alpha^s)^2 + (\beta_k - \beta^s)^2]^{(1/2)}$  be the distance between the sensor  $\mathbf{s}$  and the target; in a logarithmic scale the target-originated measurements are modeled by

$$\begin{aligned} \mathbf{H}_k(\mathbf{x}_k) &= K - 10\alpha \log(\mathbf{d}_k), \\ \mathbf{z}_k &= \mathbf{H}_k(\mathbf{x}_k) + \mathbf{B}_k \mathbf{v}_k, \end{aligned} \quad (27)$$

where the measurement noise  $\mathbf{v}_k$  accounts for the shadowing effects and other uncertainties. The noise  $\mathbf{v}_k$  is assumed to be a zero-mean Gaussian with covariances values  $\sigma_x^2 = \sigma_y^2 = \sigma_o^2$ , and the sensor noises are assumed uncorrelated;  $K$  is the transmission power, and  $\alpha \in [2, 5]$  is the path loss exponent.

Consequently, a straightforward calculation of (15) gives

$$\begin{aligned} \mathbf{D}_k^{11} &= \mathbf{F}_k^T \mathbf{Q}_k^{-1} \mathbf{F}_k, \\ \mathbf{D}_k^{12} &= -\mathbf{F}_k^T \mathbf{Q}_k^{-1}, \\ \mathbf{D}_k^{22} &= \mathbf{Q}_k^{-1} + E \left\{ \nabla_{\mathbf{x}_{k+1}} \mathbf{H}_{k+1}^T(\mathbf{x}_{k+1}) \mathbf{R}_{k+1}^{-1} \cdot \nabla_{\mathbf{x}_{k+1}}^T \mathbf{H}_{k+1}(\mathbf{x}_{k+1}) \right\}. \end{aligned} \quad (28)$$

From (11), the recursion of information matrix can be written as

$$\mathbf{J}_{k+1} = \left( \mathbf{Q}_k + \mathbf{F}_k \mathbf{J}_k^{-1} \mathbf{F}_k^T \right)^T + E \left\{ \nabla_{\mathbf{x}_{k+1}} \mathbf{H}_{k+1}^T \mathbf{R}_{k+1}^{-1} \nabla_{\mathbf{x}_{k+1}}^T \mathbf{H}_{k+1} \right\}, \quad (29)$$

where

$$\nabla_{\mathbf{x}_k} \mathbf{H}_k = -10\alpha \begin{bmatrix} \frac{\alpha_k}{\sqrt{\|\mathbf{x}_k - \mathbf{s}_1\|}} & 0 & \frac{\beta_k}{\sqrt{\|\mathbf{x}_k - \mathbf{s}_1\|}} & 0 \\ \frac{\alpha_k}{\sqrt{\|\mathbf{x}_k - \mathbf{s}_2\|}} & 0 & \frac{\beta_k}{\sqrt{\|\mathbf{x}_k - \mathbf{s}_2\|}} & 0 \\ \dots & \dots & \dots & \dots \\ \frac{\alpha_k}{\sqrt{\|\mathbf{x}_k - \mathbf{s}_m\|}} & 0 & \frac{\beta_k}{\sqrt{\|\mathbf{x}_k - \mathbf{s}_m\|}} & 0 \end{bmatrix}. \quad (30)$$

Substituting (30) in (29) the recursion of information matrix is given by

$$\mathbf{J}_{k+1} = \left( \mathbf{Q}_k + \mathbf{F}_k \mathbf{J}_k^{-1} \mathbf{F}_k^T \right)^{-1} + \frac{100\alpha^2}{\sigma_o} \begin{bmatrix} m\alpha_k^2 & \alpha_k^m \beta_k^m & & \\ \frac{\sum_{i=1}^m \|\mathbf{x}_k - \mathbf{s}_i\|^2}{\sum_{i=1}^m \|\mathbf{x}_k - \mathbf{s}_i\|^2} & 0 & \frac{\alpha_k^m \beta_k^m}{\sum_{i=1}^m \|\mathbf{x}_k - \mathbf{s}_i\|^2} & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\alpha_k^m \beta_k^m}{\sum_{i=1}^m \|\mathbf{x}_k - \mathbf{s}_i\|^2} & 0 & \frac{m\beta_k^2}{\sum_{i=1}^m \|\mathbf{x}_k - \mathbf{s}_i\|^2} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (31)$$

The initial information matrix required for the recursion is calculated from the prior probability density function  $p(\mathbf{x}_0)$ . We assume that the initial target state is a Gaussian random variable and  $\mathbf{x}_0 \sim \mathcal{N}(\hat{\mathbf{x}}_0, \mathbf{P}_0)$ . Then, if the state is Gaussian with covariance  $\mathbf{P}_0$ ,  $\mathbf{J}_k$  can be recursively computed from the initial condition  $\mathbf{J}_0 = \mathbf{P}_0^{-1}$ .

**6.3. Tracking Accuracy.** We implemented the node's selection algorithms and the particle filters in a Matlab simulator. We present simulation results for the scenarios illustrated in Table 3. Figure 3 shows the target trajectories for two different velocities, equal to 0.1 and 0.3 m/s, used in the experiment.

In Figures 4 and 5, we have shown the PCRBs for the position, meaning that the (1,1) and (3,3) elements of  $\mathbf{J}^{-1}$  are considered. Furthermore, we compare the PCRb with the root mean square errors of PF and UPF for different numbers  $m$  of active nodes. It should be noted that the empirical error curves for the PF and the UPF closely match the theoretical PCRb for the problem considered. This means that the PF and the UPF appear to be efficient sequential estimators of the target state vector. The simulation results provide that the PCRb decreases as the number of active nodes increases. Note that in Figure 4(d), contrarily to the PCRb, the root mean square error of the two filters shows a divergence from the expected decreasing behavior. We believe it is due to the degeneracy phenomenon of particle filters; however, an additional investigation is needed.

Figure 6 shows the root-mean-squared error (RMSE) on the position of the target of different filters versus the number of desired nodes using 100 runs. The bootstrap particle filter

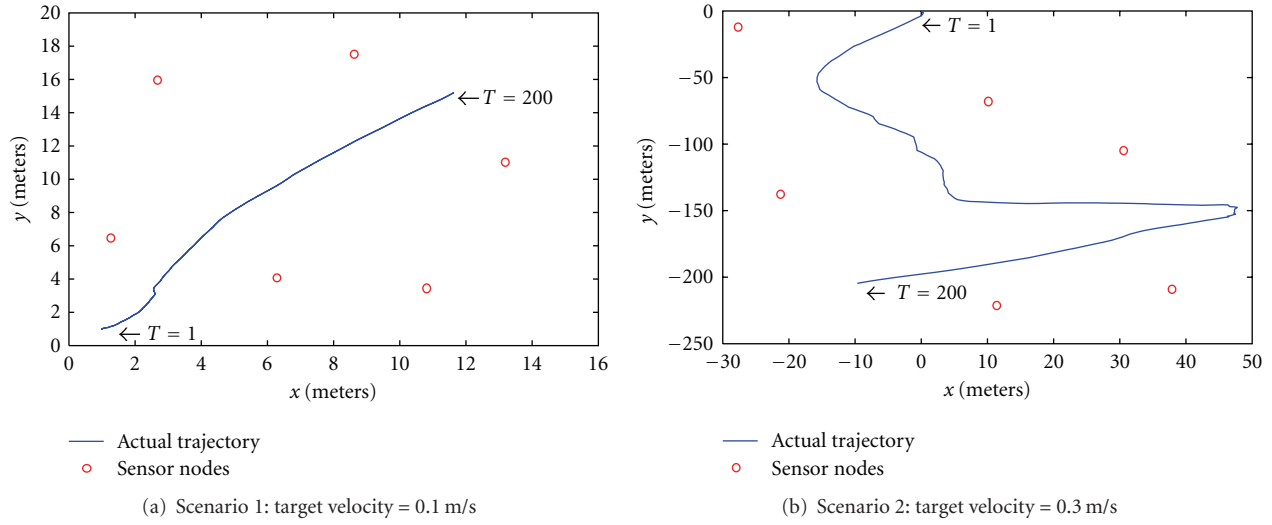


FIGURE 3: Actual track of the target with different velocity.

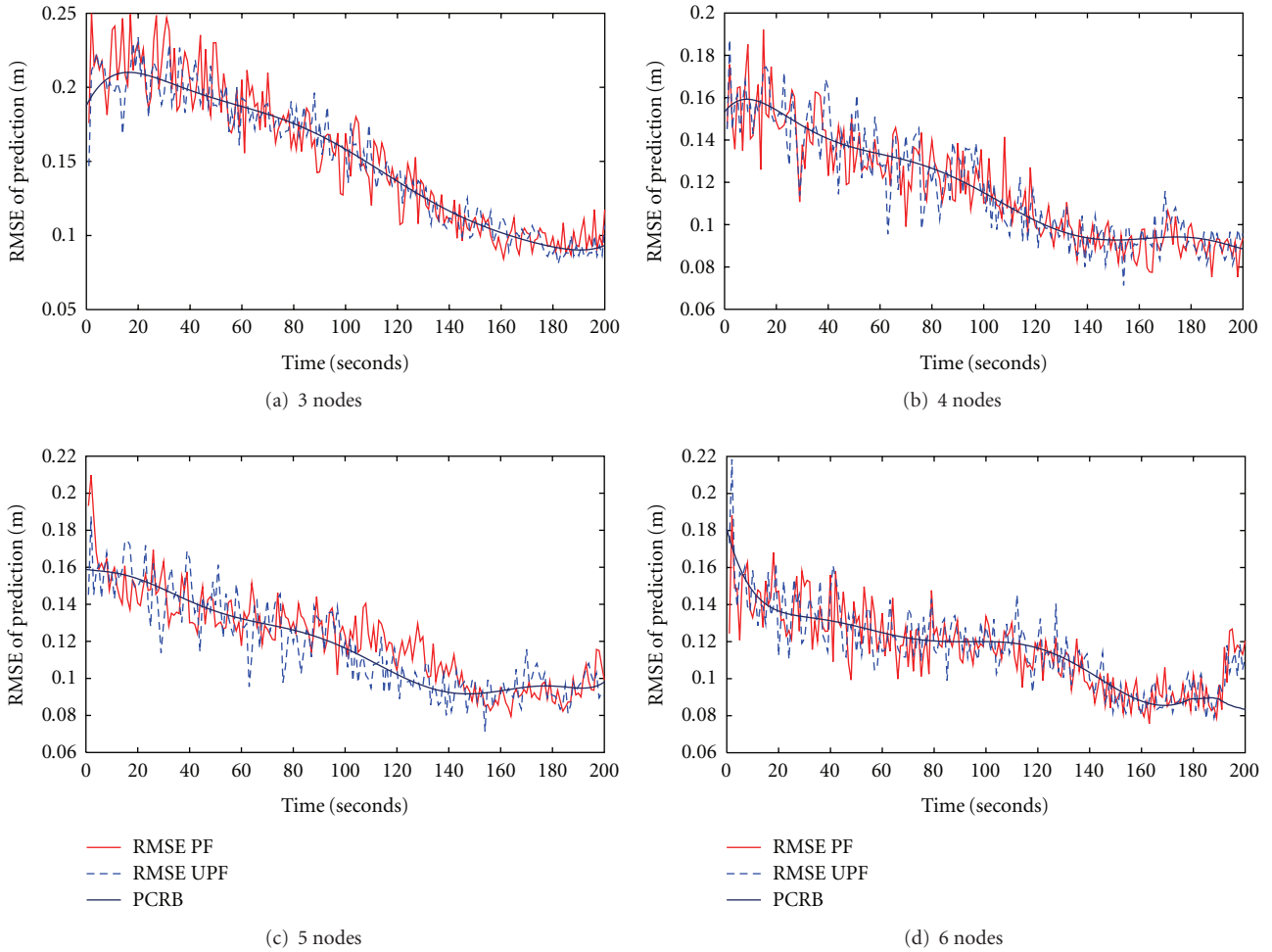


FIGURE 4: Compared tracking error versus time with target velocity = 0.1 m/s.

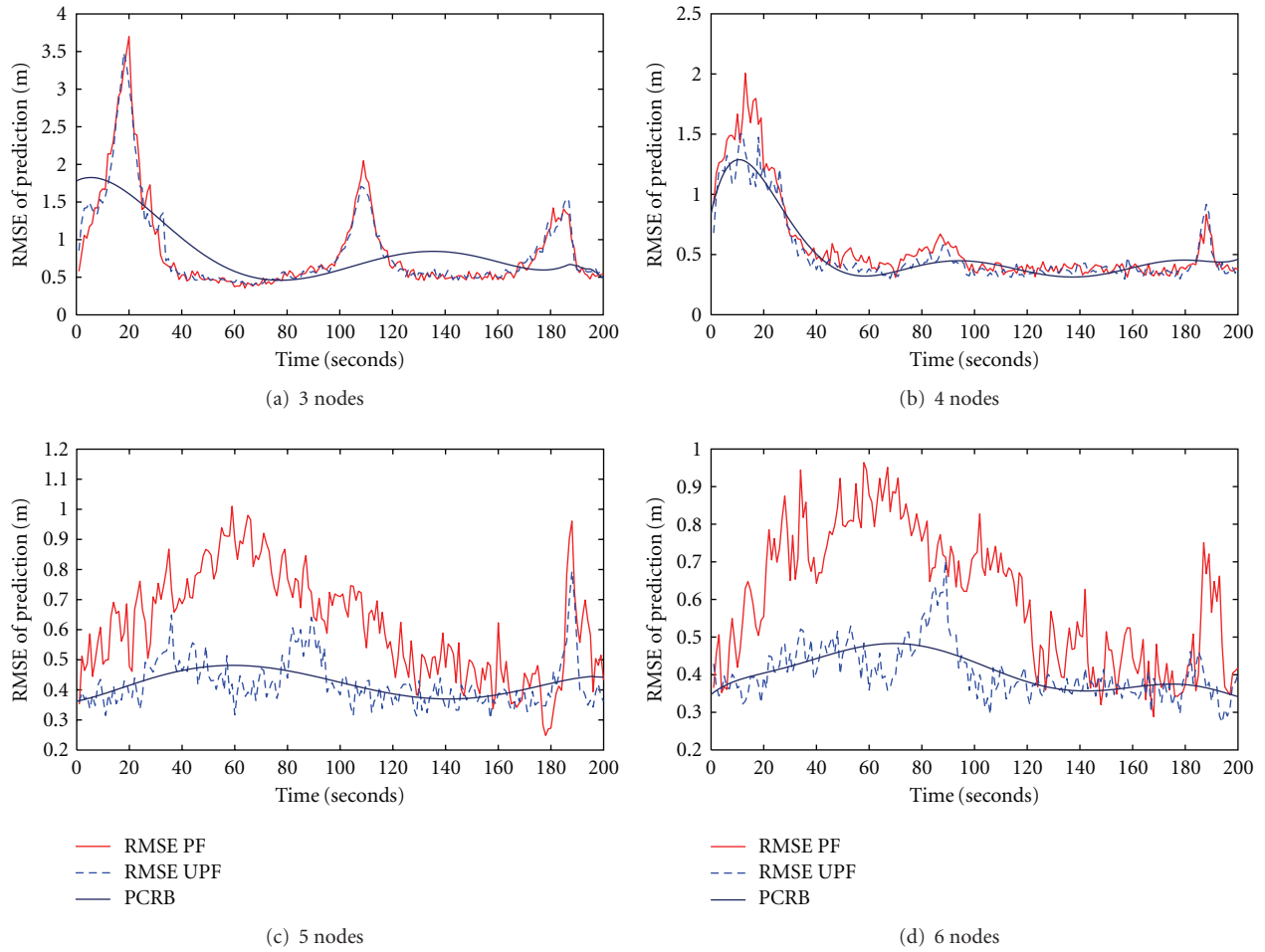


FIGURE 5: Compared tracking error versus time with target velocity = 0.3 m/s.

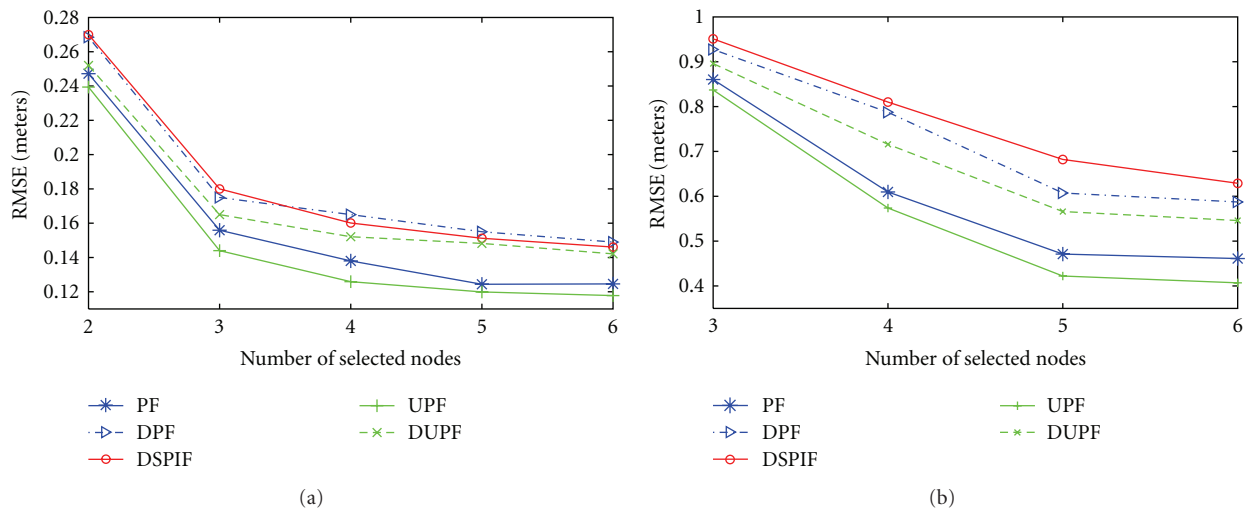


FIGURE 6: Performance accuracy comparison of particle filters: (a) target velocity = 0.1 m/s,  $\sigma_o = 0.3$ , and  $\sigma_p = 1.0$ ; (b) target velocity = 0.3 m/s,  $\sigma_o = 0.3$ , and  $\sigma_p = 1.8$ .

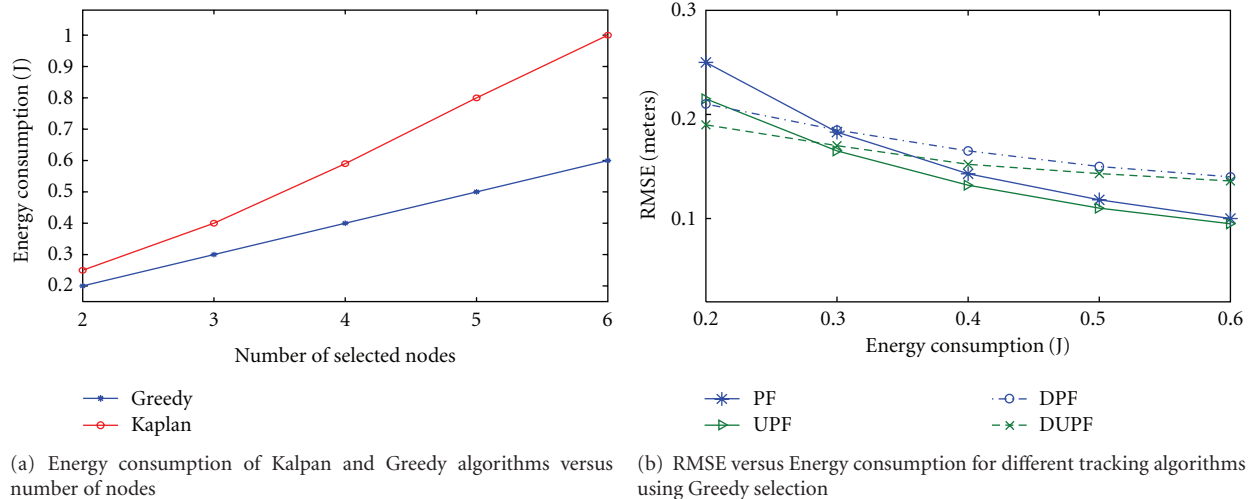


FIGURE 7: Energy consumption comparison.

and the unscented particle filter have been implemented in both centralized and distributed manner using the node selection rules. The performance of the distributed PF (DPF) and distributed UPF (DUPF) is compared with the performance of the distributed sigma-point information filter (DSPIF) from [4]. Confidence intervals are not shown for the sake of clarity.

In Scenario1, nodes are randomly deployed on an area of  $20\text{ m} \times 20\text{ m}$  and the target speed is  $0.1\text{ m/s}$  (see Figure 3(a)). In Figure 6(a) a process variance  $\sigma_p$  equal to 1.0 has been used for the trajectory 3(a). Clearly, the centralized filter PF outperforms the distributed filter DPF in tracking quality because in the distributed algorithms nodes only have local knowledge. Also the RMSE of DUPF is always larger than that for UPF. On the other hand, as we will show, the energy consumption is higher for the centralized approach. Finally, DPF and DUPF outperform DSPIF.

In Scenario2, nodes are randomly deployed on an area of  $200\text{ m} \times 200\text{ m}$  and the target speed is  $0.3\text{ m/s}$  (see Figure 3(b)). Figure 6(b) illustrates the root-mean-squared error (RMSE) on the position of the target, depending on the number of active nodes in the network. Figure 6(b) shows the RMSE for a process variance  $\sigma_p$  equal to 1.8. Again the unscented particle filter with 100 particles gives best results than the bootstrap particle filter using 100 particles as it is clear in Figure 6(b). Simulation results indicate a decrease in tracking performance with increase of noise and fast target movement. Note that, in Figure 6(b), the values of the error when the number of desired nodes is equal two are omitted because in this case the filter diverges. Other simulation results that we have not reported, with target velocity equal to 0.5 and  $1.0\text{ m/s}$ , show that to estimate the track when the velocity increases a high number of anchor nodes are needed. For each value of  $N_d$ , 10000 different random configurations were generated, where, for each configuration, we assume a maximum range between node and target equal to 30 meters, and a maximum range between node and cluster head equal to 10 meters. The DPF and DUPF computational complexity is given by  $O(N^3)$ , while

the *Kalpan* computational complexity is given by  $O(N^2)$  as the Kalman filter has been used. Particularly, the time to process the bootstrap particle filter with 100 and 500 particles is equal to 1,605 seconds and 8,052 seconds, respectively, with three active nodes, while the time to process the unscented particle filter with 100 particles is equal to 3,281 seconds. In conclusion, the UPF is less computational efficient than the PF but performs a more accurate estimation of the target's position compared to the UPF.

**6.4. Energy Consumption.** Figure 7 shows the energy consumption of node selection algorithms. In Figure 7(a), we compare the energy consumption of the proposed greedy algorithm and Kaplan algorithm as the number of selected nodes increases, using the the residual energy-based metrics defined in (3). Simulation results indicate an increase of the energy consumption with growing number of nodes. We highlight that the rise of the greedy algorithm energy consumption is superlinear using the energy-based metric introduced by Heinzelman in [10] while the rise is linear using our proposed residual energy-based metric. Definitely can be concluded that the greedy selection algorithm outperforms the Kaplan selection algorithm to select the sensors that would give the most prolonged life to the network. In Figure 7(b), RMSE versus energy consumption of PF, DPF, UPF, and DUPF algorithms using greedy selection has been shown.

In conclusion, the energy consumption increases with the number of active nodes; on the other hand the tracking error decreases as the the number of active nodes increases. A tradeoff between the performance and the number of nodes is needed to save energy.

## 7. Conclusion

The focus of the article was the energy-efficient and collaborative target tracking in wireless sensor networks. The tracking problem was formulated as a cross-layer optimization with the aim of maximizing the total utility



function in the cluster. The node selection procedures were integrated into a particle filter and tested on simulated data. The optimal greedy-based algorithm was extended at the real scenario to consider the energy consumption over more measurement intervals. A lower bound was also introduced. The experiments indicate that the proposed approach outperforms the existing algorithms in literature. Extensive simulations showed that the target tracking system yields good accuracy for lower velocity of the target. The tracking performances get worse as the noise and the target velocity increase. The bootstrap particle filter and the unscented particle filter for the centralized and distributed scheme and the distributed sigma-point information filter [4] have been implemented and the accuracies have been compared. Furthermore, we have presented a closed-form derivation of PCRB as a performance criterion, eliciting the influence of the number of active nodes, the channel parameters and the model parameters. The formula has been derived and the tracking protocol has been evaluated for linear dynamic's models and for a nonlinear Gaussian observation's model. Finally, in this study we assumed an intra-cluster communication and limited ourselves to consider a single cluster in which an even energy consumption by the sensor nodes has been assumed. Future work is investigating the implementation of algorithms to report tracking samples to multiple cluster heads.

## Acknowledgement

The author would like to thank Professor Ian Akyildiz for the helpful suggestions.

## References

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393–422, 2002.
- [2] A. T. Ihler, J. W. Fisher III, R. L. Moses, and A. S. Willsky, "Nonparametric belief propagation for self-localization of sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 809–819, 2005.
- [3] J. Lee, K. Cho, S. Lee, T. Kwon, and Y. Choi, "Distributed and energy-efficient target localization and tracking in wireless sensor networks," *Computer Communications*, vol. 29, pp. 2494–2505, 2006.
- [4] T. Vercauteren and X. Wang, "Decentralized sigma-point information filters for target tracking in collaborative sensor networks," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2997–3009, 2005.
- [5] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," in *Proceedings of 5th ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '99)*, pp. 263–270, Seattle, Washington, August 1999.
- [6] E. Ertin, J. W. Fisher, and L. C. Potter, "Maximum mutual information principle for dynamic sensor query problems," in *Proceedings of the International Workshop on Information Processing in Sensor Networks (IPSN '03)*, pp. 405–416, Palo Alto, Calif, USA, April 2003.
- [7] H. Wang, G. Pottie, K. Yao, and D. Estrin, "Entropy-based sensor selection heuristic for target localization," in *Proceeding of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pp. 36–45, Berkeley, Calif, USA, April 2004.
- [8] L. M. Kaplan, "Global node selection for localization in a distributed sensor network," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 113–135, 2006.
- [9] L. M. Kaplan, "Local node selection for localization in a distributed sensor network," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 136–146, 2006.
- [10] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," in *Proceedings of 33rd Hawaii International Conference on System Sciences (HICSS '00)*, Maui, Hawaii, January 2000.
- [11] L. Arienzo and M. Longo, "Energy-efficient tracking strategy for wireless sensor networks," in *Proceeding of the 5th IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS '08)*, pp. 595–602, Atlanta, Ga, USA, September 2008.
- [12] H. T. Kung and D. Vlah, "Efficient location tracking using sensor networks," in *Proceedings of the IEEE Wireless Communications & Networking Conference (WCNC '03)*, March 2003.
- [13] C.-Y. Lin and Y.-C. Tseng, "Structures for in-network moving object tracking in wireless sensor networks," in *Proceeding of the 1st International Conference on Broadband Networks (BroadNets '04)*, pp. 718–727, October 2004.
- [14] W. Zhang and G. Cao, "DCTC: dynamic convoy tree-based collaboration for target tracking in sensor networks," *IEEE Transactions on Wireless Communications*, vol. 3, no. 5, pp. 1689–1701, 2004.
- [15] W. Zhang and G. Cao, "Optimizing tree reconfiguration for mobile target tracking in sensor networks," in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, vol. 4, pp. 2434–2445, 2004.
- [16] Q. Wang, W.-P. Chen, R. Zheng, K. Lee, and L. Sha, "Acoustic target tracking using tiny wireless sensor devices," in *Proceedings of the International Workshop on Information Processing in Sensor Networks (IPSN '03)*, pp. 642–657, 2003.
- [17] W.-P. Chen, J. C. Hou, and L. Sha, "Dynamic clustering for acoustic target tracking in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 258–271, 2004.
- [18] J. Liu, J. Reich, and F. Zhao, "Collaborative in-network processing for target tracking," *EURASIP Journal on Applied Signal Processing*, vol. 2003, no. 4, pp. 378–391, 2003.
- [19] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration for tracking applications," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 61–72, 2002.
- [20] C. M. Kreucher, A. O. Hero, K. D. Kastella, and M. R. Morelande, "An Information-based approach to sensor management in large dynamic networks," *Proceeding of IEEE*, vol. 95, pp. 978–999, 2007.
- [21] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.
- [22] J. H. Chang and L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks," in *Proceedings of the 19th Annual Conference on Computer Communications (INFOCOM '00)*, vol. 1, pp. 22–31, Tel Aviv, Israel, March 2000.

- [23] I. Stojmenovic and X. Lin, "Power-aware localized routing in wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 11, pp. 1122–1133, 2001.
- [24] H. Van Trees, *Detection, Estimation, and Modulation Theory, Part I*, John Wiley & Sons, New York, NY, USA, 1968.
- [25] P. Tichavsky, C. H. Muravchik, and A. Nehorai, "Posterior cramer-rao bounds for discrete-time nonlinear filtering," *IEEE Transactions on Signal Processing*, vol. 46, no. 5, pp. 1386–1396, 1998.
- [26] A. Doucet, N. De Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, Springer, London, UK, 2001.
- [27] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings-F*, vol. 140, no. 2, pp. 107–113, 1993.
- [28] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.
- [29] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, John Wiley & Sons, New York, NY, USA, 2001.
- [30] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*, Academic Press, New York, NY, USA, 1970.
- [31] R. Van der Merwe, A. Doucet, N. De Freitas, and E. Wan, "The unscented particle filter," in *Advances in Neural Information Processing Systems*, pp. 548–590, 2000.
- [32] D. Guo and X. Wang, "Dynamic sensor collaboration via sequential Monte Carlo," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 1037–1047, 2004.
- [33] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1982.
- [34] T. A. Feo and M. G. C. Resende, "Greedy randomized adaptive search procedures," *Journal of Global Optimization*, vol. 6, no. 2, pp. 109–133, 1995.
- [35] W. L. G. Koontz, P. M. Narendra, and K. Fukunaga, "A branch and bound clustering algorithm," *IEEE Transactions on Computers*, vol. 24, no. 9, pp. 908–915, 1975.
- [36] L. Arienzo, *Energy-efficient target tracking through wireless sensor networks. Cross-layer design and optimization*, Ph.D. thesis, School of Electrical and Information Engineering, University of Salerno, Salerno, Italy, March 2008.
- [37] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall, Englewood Cliffs, NJ, USA, 2nd edition, 2001.