Energy-Efficient Target Tracking in Sensor Networks

Loredana Arienzo[1,2] and Maurizio Longo[2]

[1] Institute for the Protection and Security of the Citizen, Joint Research Centre, European Commission
Ispra, 21027, Varese, Italy
loredana.arienzo@jrc.it
[2] Department of Electrical and Information Engineering, University of Salerno
Fisciano, 84084, Salerno, Italy
longo@unisa.it

Abstract. In this paper, the problem of collaborative tracking of mobile nodes in wireless sensor networks is addressed. Aiming at an energy efficient solution, we propose 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 using the energy model by Heinzelman, 2000. We layout a cluster-based architecture to address the limitations in computational, battery power and communications of the sensor devices. The node selection problem is formulated as a cross-layer optimization problem that is solved using a greedy algorithm. 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 performance are compared with the distributed sigma-point information filter in literature, under two common channel models: the log-normal shadowing and the Rayleigh fading.

Keywords: Sensor networks, tracking, particle filters, energy efficient, cross-layer optimization, cluster.

1 Introduction

The issue of tracking moving targets in wireless sensor networks (WSNs) [1] has received significant attention in recent years [2–4]. A tracking algorithm designed for sensor networks 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.

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

J. Zheng, D. Simplot-Ryl, V.C.M. Leung (Eds.): ADHOCNETS 2010, LNICST 49, pp. 249–264, 2010.
© Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2010

nodes for tracking a target in a distributed wireless sensor network was investigated since 1999 [5]. The main idea is for a network to determine participants in a *sensor collaboration* by dynamically optimizing an utility function of data for a given cost of communication and computation. Previous research [6–9], has focused on information-theoretic node selection approaches, i.e. on heuristics to select an informative sensor such that the fusion of the selected sensor observation with the prior target location distribution would yield 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 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.

Many criteria influence the design of energy-efficient tracking approaches, and a wide range of schemes have been proposed. The rest of this Section provides a partial overview of such schemes. Wang *et al.* [10], Chen *et al.* [11], propose *cluster-based* tracking schemes. They envision a hierarchical network composed of (a) a static backbone of sparsely placed position-aware sensors which may 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 proximity), and a CH collects information from its cluster members and determines the target location using either the trilateration technique [10] or the Voronoi diagram-based approach [11]. 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 a *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 [12–14] that use a global information, the cluster-based schemes [10, 11] 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 are centralized approaches, while the cluster-based are distributed schemes in which the CH is the *leader node* in the processing. References [13, 15] are centralized approaches, while [3, 4, 10, 14, 16] are distributed approaches. As defined in [17] the *sensor management* is the process of dynamically retasking sensors in response to an evolving environment. In a tracking task the sensor 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 as information-driven or information-based. Zhao *et al.* [16] 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 these approaches is to optimize an utility function, representing the location accuracy, using entropy-based metrics.

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 optimize an utility function combining the overall energy in a cluster and the mutual information between the sensor node observations and the target state. The remainder of the paper is organized as follows: Section 2 describes the model of the overall system by introducing the network architecture and energy-based metrics. Section 3 provides the model of the dynamical system being sensed. 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. Section 6 discusses the performances of proposed algorithms, while in Section 7, we draw the main conclusions.

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 that are candidate to assume the role of a CH and 2) randomly distributed low-end sensors which sense a moving target and report data to their CH upon request. Finally, we assume that the network is composed of dynamic clusters, depending on the predicted target trajectory (see Fig. 1). 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. The acting CH will predict the trajectory of a target by means of a particle filter based on the history of the target location and some recent observations communicated by a subset of active sensors in the cluster. This subset is selected to minimize the overall energy consumption.

2.1 Energy Model

To describe the energy consumption of a tracking algorithm, we use the energy model for wireless sensor networks introduced by Heinzelman *et al.* [18]. Hence, given l bits of data, the overall energy consumption to transmit the packet of l bits between two nodes at a distance d with a given received SNR can be expressed as:

$$E(d, l) = (2E_{elec} + E_{amp} \cdot d^{\alpha}) \cdot l \tag{1}$$

where E_{elec} $[Joule/bit]$ is the energy needed by the transceiver circuitry to transmit or receive one bit and E_{amp} $[Joule/(bit \cdot m^{\alpha})]$ is a constant which represents

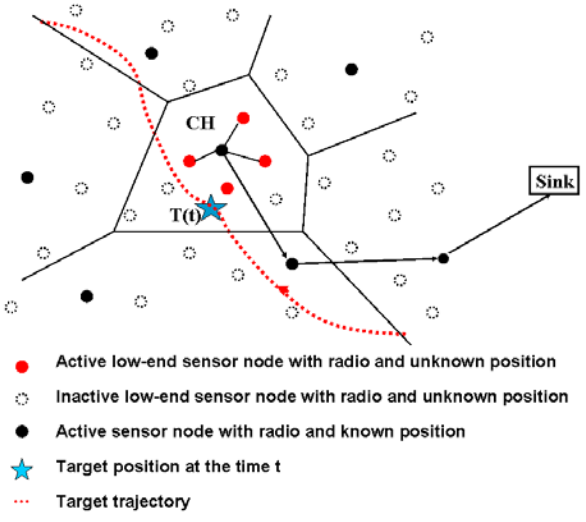


Fig. 1. Sensor Network Topology

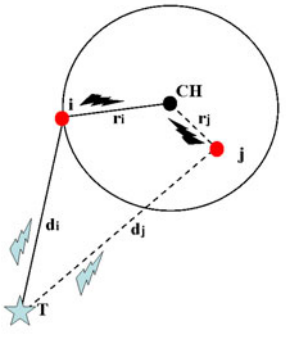


Fig. 2. Location Discovery Protocol

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 (1) as the energy-based metric.

In our analysis we adopt a slightly different metric accounting not only for the energy consumption as in the above model but also for the remaining energy at the node. Hence, if we refer to a link (i,j) with distance d_{ij} along which a l_{ij}-bit packets is transmitted, the residual energy at node i evolves in time according to:

$$E_r(i,k+1) = E_r(i,k) - \sum_j (2E_{elec} + E_{amp} \cdot d_{i,j}^{\alpha}(k)) \cdot l_{i,j} \qquad (2)$$

where $E_r(i,k)$ is the remaining energy at node i and time k. Note that, in the scenario of a mobile node i, d_{ij} is a function of time k. We refer to (2) as the residual energy-based metric.

2.2 Network Model

As stated before, we consider a network composed of randomly deployed sensor nodes which sense a moving target and transmit their observations to a position-aware sensor which acts as CH, namely the data gathering node. The network is divided into N_c clusters each having N_a neighbors, i.e. sensor nodes within the radio range of the cluster head. Each sensor is equipped with a low data rate radio interface. The CHs are equipped with two radio transmitters, i.e., a low data rate transmitter to communicate with the sensors, and a high rate wireless interface for CH-CH communication.

We assume that the (static) position of the CHs is known and want to estimate the distance of each neighbor with respect to the acting 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, as in [4], where sensors measure the power of a radio signal emitted by the object: assuming the exponential decrease of received power with distance and the log-normal shadowing model for the observations we get

$$P_r(d) = P_r(d_o) \cdot (d_o/d)^{\alpha} + X_\sigma \tag{3}$$

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, α is the path loss exponent $\alpha \in [2,5]$, and X_σ is the shadow fading component, with X_σ 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 strengthat the CH at , P_a is the strength of the signal emitted by the sensor.

3 Model for State Estimation of a Dynamical System

The problem of estimating the state vector of a 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 obey the equations

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{A}_k \mathbf{w}_k \tag{4}$$

$$\mathbf{z}_k = \mathbf{H}_k(\mathbf{x}_k) + \mathbf{B}_k \mathbf{v}_k \tag{5}$$

where, k is the discrete time index; $\mathbf{x}_k \epsilon \mathbf{R}^n$ is the state vector; $\mathbf{z}_k \epsilon \mathbf{R}^m$ is the observation vector; $\mathbf{w}_k$ is a zero-mean Gaussian process noise with nonsingular covariance matrix $\mathbf{Q}_k$; $\mathbf{v}_k$ is a zero-mean Gaussian measurement noise independent of $\mathbf{w}_k$ with nonsingular covariance matrix $\mathbf{R}_k$. The matrices $\mathbf{F}_k$, $\mathbf{A}_k$, $\mathbf{B}_k$ are

independent of the state vector whereas $\mathbf{H}_k$ is a function dependent of the state vector $\mathbf{x}_k$. Further we assume that the initial state $\mathbf{x}_0$ has a known probability density function $p(\mathbf{x}_0)$.

In this paper we assume as dynamic model of the target the constant velocity model. 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 given by,

$$\mathbf{x}_{k+1} = \begin{pmatrix} 1 & \Delta\mathrm{T} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta\mathrm{T} \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{x}_k + \begin{pmatrix} \Delta\mathrm{T}^2/2 & 0 \\ \Delta\mathrm{T} & 0 \\ 0 & \Delta\mathrm{T}^2/2 \\ 0 & \Delta\mathrm{T} \end{pmatrix} \mathbf{w}_k$$

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

3.1 Observation Model for Lognormal Shadowing

As observation model of the measurements, we first use the log-normal shadowing model [19]. Hence, let $\{\alpha^s, \beta^s\}$ be the fixed position of sensor $\mathbf{s}$ and $\mathbf{d}_k = \|\mathbf{x}_k - \mathbf{s}\|^{1/2} = [(\alpha_k - \alpha^s)^2 + (\beta_k - \beta^s)^2]^{(1/2)}$ be the distance between the sensor $\mathbf{s}$ and the target, in logarithmic scale the measurements are modeled by

$$\mathbf{H}_k(\mathbf{x}_k) = K - 10\alpha \log(\mathbf{d}_k) \tag{6}$$

$$\mathbf{z}_k = \mathbf{H}_k(\mathbf{x}_k) + \mathbf{B}_k \mathbf{v}_k \tag{7}$$

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

3.2 Observation Model for Rayleigh Fading

The problem of estimating the state vector of a nonlinear dynamical system over a fading channel can be formulated as follows. The state and the observations of the target of interest are assumed to obey the equations

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{A}_k \mathbf{w}_k \tag{8}$$

$$\mathbf{z}_k = \mathbf{H}_k(\mathbf{x}_k)\mathbf{r}_k + \mathbf{B}_k \mathbf{v}_k \tag{9}$$

where $\mathbf{r}_k$ is the fading channel gain. Specifically, for a Rayleigh fading channel, the probability density of $\mathbf{r}_k$ is:

$$p_R(\mathbf{r}_k) = \frac{\mathbf{r}_k}{\sigma_R^2} \exp(-\frac{\mathbf{r}_k^2}{2\sigma_R^2}), \quad \mathbf{r}_k \geq 0. \tag{10}$$

4 Tracking Algorithms

As to the tracking task, we use sequential Monte Carlo (SMC) approaches, also known as particle filtering [20, 21]. A different approach by 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 main idea of particle filtering (PF) 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 sequential importance sampling (SIS). A proposal or importance distribution is introduced, which is an approximation of the required posterior distribution of states given the value of previous step $\pi(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_1, ..., \mathbf{z}_{k-1})$.

As the number of samples becomes very large, the SIS filter approaches the optimal Bayesian estimate. A common problem with the SIS PF is the degeneracy phenomenon, since after a few iterations all the particles, except one, 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 PF using sequential importance resampling (SIR) techniques is known as *bootstrap filter* or SIR PF. // 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 mitigate the degeneracy problem. Notable techniques include local linearization using the extended Kalman filter (EKF) or the unscented Kalman filter (UKF) to estimate the importance distribution. A particle filter which uses UKF to generate the importance distribution is referred as *unscented particle filter* (UPF) or sigma-point particle filter [22]. In particular, the unscented transformation is used to generate and propagate a Gaussian proposal distribution for each particle. The unscented transformation still approximates the proposal distribution by a Gaussian distribution, but it is specified using a minimal set of deterministically chosen sample points or sigma points. These points complectly capture the true mean and covariance of the Gaussian distribution, and when propagated through the true nonlinear system, they accurately capture the posterior mean and covariance to the third order for any nonlinearity.

In this paper, we implement both the SIR PF and the UPF and we compare thire performances in terms of estimation accuracy and energy efficiency when integrated into collaborative and distributed schemes for tracking a moving target.

4.1 Node Selection by Information

According to SMC approach, each new sensor measurement $\mathbf{z}_k$ is combined with the current estimate $p(\mathbf{x}_k|\mathbf{z}_1, ..., \mathbf{z}_{k-1})$, hereafter called belief state, to form a

new belief state $p(\mathbf{x}_k|\mathbf{z}_1, ..., \mathbf{z}_k)$. We consider now the problem of selecting a new sensor in order to provide the greatest improvement of accuracy at the lowest cost. Let Z_k represent all measurements that have already been used at time k in the inference of the current belief state, which is maintained at the CH. The objective function for this optimization problem can be defined as a mixture of both information gain and cost. In the remainder of this Section we consider the information gain, while the energy consumption issue is discussed in next Section.

The information gain stemming from selection of sensor s can be defined as $\tilde{\Phi}_s(p(\mathbf{x}_k|Z_k)) = \Phi_{Utility}(p(\mathbf{x}_{k+1}|Z_k, \mathbf{z}_{k+1,s}))$, 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], i.e. $\tilde{\Phi}_s(p(\mathbf{x}_k|Z_k)) = H_{target}(Z_k) - H_{target}(Z_k, \mathbf{z}_{k+1,s})$; or, equivalently as the mutual information $\Phi_{Utility}(p(\mathbf{x}_{k+1}, \mathbf{z}_{k+1,s}|Z_k)) = I(\mathbf{x}_{k+1}, \mathbf{z}_{k+1,s}|Z_k)$ conveyed on $\mathbf{x}_{k+1}$ by the new measurements $\mathbf{z}_{k+1,s}$ [15]. 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 *aposteriori* 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 $p(\mathbf{x}_{k+1}|\mathbf{z}_{k+1,s}, Z_k) \propto p(\mathbf{z}_{k+1,s}|\mathbf{x}_{k+1})p(\mathbf{x}_{k+1}|Z_k)$. Since $\mathbf{z}_{k+1,s}$ is not available, we may 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 $\hat{p}(\bar{\mathbf{z}}_{k+1,s}|\mathbf{x}_{k+1}) = \int p(\mathbf{z}_{k+1,s}(\nu_{k+1})|\mathbf{x}_{k+1}) \times p(\nu_{k+1}|Z_k)d\nu_{k+1}$. Then, the expected posterior belief can be defined as follows:

$$\hat{p}(\mathbf{x}_{k+1}|\bar{\mathbf{z}}_{k+1,s}, Z_k) = \hat{p}(\bar{\mathbf{z}}_{k+1,s}|\mathbf{x}_{k+1})p(\mathbf{x}_{k+1}|Z_k). \tag{11}$$

The entropy of expected posterior distribution can be computed based on the discrete belief state $\{\mathbf{x}_k^j, w_k^j\}_{j=1}^L$ [23], where w_k^j is the importance sampling weight in the resampling step of the particle filters and L represents the number of weights, i.e. the number of particles. According to (11), the expected posterior belief for sensor s can be represented by the discrete belief state $\{\mathbf{x}_k^j, \tilde{w}_{k+1,s}^j\}_{j=1}^L$ with the weights $\tilde{w}_{k+1,s}^j$ given by

$$\tilde{w}_{k+1,s}^j = \hat{p}(\bar{\mathbf{z}}_{k+1,s}|\mathbf{x}_{k+1}^j)w_{k+1}^j. \tag{12}$$

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

$$H = -\sum_{j=1}^{L} \tilde{w}_{k+1,s}^j \log \tilde{w}_{k+1,s}^j. \tag{13}$$

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

$$\begin{aligned}\mathcal{N}_s &= arg \max_{s \epsilon \mathcal{N}_a} \tilde{\Phi}_s(p(\mathbf{x}_k|Z_k)) \\ &= arg \max_{s \epsilon \mathcal{N}_a} (H_{target}(Z_k) - H_{target}(Z_k, \mathbf{z}_{k+1,s}))\end{aligned} \tag{14}$$

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

5 Energy Efficient Tracking

Let us consider the following location discovery protocol for a given snapshot. Let $\mathcal{N}_s$, as above, be the set of active neighbor nodes maximizing the utility function as in (14). We now want to select, inside $\mathcal{N}_s$, a subset $\mathcal{N}_d$ of N_d active nodes collaborating in the localization task with an aim to minimize the energy consumption. 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.

According to metric (1), and assuming that target T emits discovery signal with period T_M, the energy cost associated with sensor $i \in \mathcal{N}_s$ is given by:

$$\begin{aligned}E_i(r_i, d_i) = &[E_{elec}(N_d + 2) + E_{amp} \cdot d_i^{\alpha}] \cdot \frac{l}{T_M} \\ &+ [E_{elec}(N_d + 1) + E_{amp} \cdot r_i^{\alpha}] \cdot b\end{aligned} \tag{15}$$

where b represents the bit rate [bit/s] between the CH and the neighbor i, r_i and d_i are, respectively, the distance of the node i from the CH and the target, $E_{elec}N_d$ 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.

The total energy consumption by all nodes in $\mathcal{N}_s$, is hence given by

$$E^{TOT}(\mathcal{N}_s) = \sum_{i \in \mathcal{N}_s} E_i(r_i, d_i) \tag{16}$$

As our objective is to select the optimal subset $\mathcal{N}_d \subset \mathcal{N}_s$, i.e. which minimizes the total energy cost in Eq. (16) subject to a constraint of the cardinality N_d of said subset, the objective function can be formally cast as follows:

$$\begin{aligned}&\mathcal{N}_d = arg \min_{\mathcal{N} \subseteq \mathcal{N}_s} E^{TOT}(\mathcal{N}) \\ &\text{subject to } \; N_d \geq 2\end{aligned} \tag{17}$$

A unique solution to this problem exists, since the objective function is strictly concave and the feasible set is convex. Following [24], the solutions of this optimization problem depend on the scenario, static or dynamic, as illustrated in the following Subsections.

Algorithm 1. Energy Efficient Tracking Algorithm

Synopsis: $[\mathcal{N}_d, \mathcal{C}, E_{tot}]$ =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_{tot}.

1. The initial leader node does the following step:
   (a) draw initial samples $\{\mathbf{x}_0^j, w_0^j = 1\}_{j=1}^L$ of the target from the prior information; (b) update the belief state $\{\mathbf{x}_1^j, w_1^j\}_{j=1}^L$ 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 $\{\mathbf{x}_2^j, w_{2,i}^j\}_{j=1}^L$ for each neighboor node i with the weights $\tilde{w}_{2,i}^j$ computed by (12); (d) compute the entropy of the expected posterior belief state $\{\mathbf{x}_2^j, w_{2,i}^j\}_{j=1}^L$ for each neighboor node i by (13) and determine the next best sensor, say b in the set $\mathcal{N}_s$.
2. $[\mathcal{N}_d, \mathcal{C}, E_{tot}, node, E_b, k]$ =Greedy($\mathcal{N}_s$,N_d)
3. Loop until time runs out:
4. Prediction and update steps of particle filtering to estimate the target's trajectory.
5. Update the candidate set $\mathcal{C}$ during the dynamic of the target.
6. $[\mathcal{N}_d, \mathcal{C}', E_{tot}]$ =BranchBound($N_d, \mathcal{C}, E_b, node$)

5.1 The Solution in the Static Scenario

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. Here we adopt the meta-heuristic GRASP: each iteration consisting 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 [24]. It provides the set of desired nodes $\mathcal{N}_d$ with the related total energy E_{tot}, the last node selected *node* with the related specific energy E_b, the set of nodes in the cluster which are candidated for the next snapshot $\mathcal{C}$.

5.2 The Solution in the Dynamic Scenario

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 Eq. (15) varies with the time and hence the energy consumption varies with time k.

For this dynamic version of the optimization problem we use the *dynamic programming*, 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 follows: the energy bound is the maximum energy consumption, namely the energy consumed by the nodes selected in the previous snapshot. The pseudo-code of an efficient implementation of our branch-and-bound approach is described in [24]. 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_{tot}. The overall tracking algorithm which combines the node's selection procedures with the particle filtering algorithm is finally outlined as Algorithm 1.

6 Performance Evaluation

In this section, we investigate the performance of the overall target tracking system looking first at the node selection algorithm, second at the estimation bound, then at the tracking algorithm and finally at the energy consumption. We present simulation results for the scenarios illustrated in Table 1. In the experiment we use the target trajectories introduced in [24] for two different velocities, equal to 0.1 and 0.3 m/s.

Table 1. Parameters of the model used for simulations

Parameters	Scenario1	Scenario2
size	$20m^2$	$200m^2$
velocity	$0.1m/s$	$0.3m/s$
K	9 dB	9 dB
α	3	3
E_{elec}	50 nJ/bit	50 nJ/bit
E_{amp}	100 pJ/bit/m^2	100 pJ/bit/m^2
E_r	100 mJ	100 mJ
T_M	2 sec	2 sec
b	10 bit/sec	10 bit/sec
l	8 bits	8 bits
ΔT	1 sec	1 sec
σ_p	1.0	1.8
σ_o	0.3	0.3
L	100, 500	100, 500

6.1 Optimal Node Selection

First we compare from a computational point of view our node selection algorithm with the *Kaplan* algorithm in [8] and [9], the difference between the last two being that in [8] a global topology knowledge is assumed and every active node reaches the entire network, while in [9] only the knowledge of the relative position to the target and the active nodes from the previous snapshot is

required. For each iteration, using N_d as number of selected node, the computational complexity of greedy algorithm is $O(N_s - N_d)$ while the computational complexity of *Kaplan* algorithm is $O((N_s - N_d)^2)$.

As stated above, the node selection procedure is combined with the maximization of the utility function. Hence in our analysis the computational complexity of the problem in (14) needs to be considered, namely $\mathrm{O}(N_a L \log L)$ where L is the number of weights. Finally, the computational complexity for all iterations to maximize the utility function in (14) is given by $\sum_{i=1}^{N_a} iL \log L$.

In Table 2 the results of a runtime measurement are illustrated, conducted on a system with AMD opteron XP processor 250, 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_s equal to 10 and L equal to 100.

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

6.2 Tracking Accuracy

We implemented the node selection algorithms and the particle filters in a Matlab simulator. Fig. 3 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 and the log-normal shadowing model. The bootstrap and the unscented particle filters have been implemented both in centralized and distributed manner using the node selection rules. The performance of the distributed PF (DPF) and distributed UPF (DUPF) are 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 20m×20m and the target speed is 0.1 m/s. Clearly, the centralized filter PF outperforms the distributed filter DPF in tracking accuracy as it may be expected since in the distributed case 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 200m×200m and the target speed is 0.3 m/s. Fig. 3 (b) illustrates RMSE on the position of the target, depending on the number of active nodes in the network. From Fig. 3 (b), the unscented particle filter with 100 particles gives best results than the bootstrap particle filter using 100 particles. Not shown in Fig. 3 (b), the error when $N_d = 2$

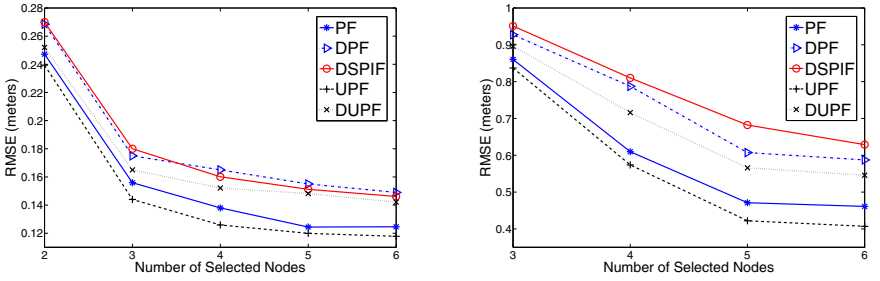


Fig. 3. Performance accuracy comparison of particle filters (a) target velocity = 0.1 m/s, $\sigma_o = 0.3$. and $\sigma_p = 1.0$, (b) target velocity = 0.3 m/s, $\sigma_o = 0.3$. and $\sigma_p = 1.8$.

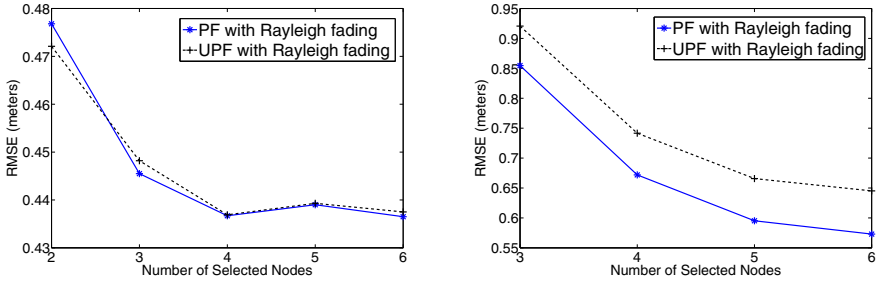


Fig. 4. Performance accuracy comparison of particle filters with fading σ_R=1.35x10^{-4}mV (a) target velocity = 0.1 m/s, (b) target velocity = 0.3 m/s

diverges. Simulation results indicate a decrease in tracking performance with increase of noise and fast target movement.

Figs. 4 and 5 show the RMSE versus N_d using 100 runs under the Rayleigh fading model. Like for the lognormal shadowing model, simulation results indicate a decrease in tracking performance as noise and target velocity increase. Note that, using 100 particles the bootstrap PF performs better than the UPF while using the lognormal shadowing model the opposite is true. This behavior could be due to the approximation of the proposal distribution that UPF uses. Indeed, while PF uses the prior distribution as proposal distribution, UPF uses a distribution depending also on observations which, in a fading channel, are characterized by fluctuations of RSS from the mean value.

For each value of N_d, 10000 different random configurations and a maximum range between node and cluster head equal to 10 meters have been considered. The DPF and DUPF computational complexity is given by O(L^3), while the *Kaplan* computational complexity is given by O(L^2). Specifically, the time to process the bootstrap particle filter with 100 and 500 particles is equal to 1,605

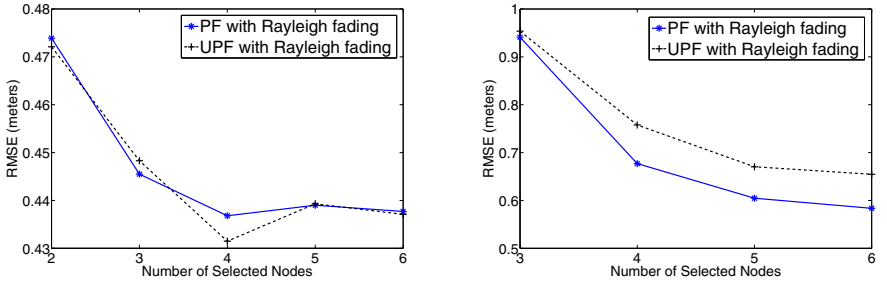
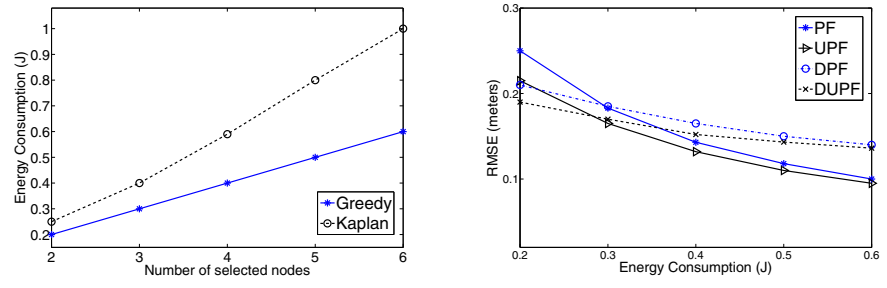


Fig. 5. Performance accuracy comparison of particle filters with fading σ_R=4.5x10^{-4}mV (a) target velocity = 0.1 m/s, (b) target velocity = 0.3 m/s



(a) Energy consumption of Kaplan and Greedy algorithms vs. number of nodes

(b) RMSE vs. Energy consumption for different tracking algorithms using Greedy selection

Fig. 6. Energy consumption comparison

sec and 8,052 sec respectively, with three active nodes, while the time to process the unscented particle filter with 100 particles is equal to 3,281 sec. 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.

6.3 Energy Consumption

Fig. 6 illustrates the energy consumption of node selection algorithms using the residual energy-based metrics defined in (2). In Fig. 6(a), the energy consumption of the greedy algorithm is compared to that of Kaplan algorithm as the number of selected nodes increases. It is very clear that the greedy selection algorithm outperforms the Kaplan selection algorithm in terms of energy efficiency. In Fig. 6(b), RMSE vs. energy consumption of PF, DPF, UPF, DUPF algorithms using greedy selection is shown. Results indicate an increase of the energy consumption with growing number of nodes.

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 Conclusions

The focus of the article was the energy-efficient and collaborative target tracking in wireless sensor networks. The tracking problem is formulated as a cross-layer optimization with an aim to minimize the total energy consumption in the cluster. The node selection procedures were integrated into several tracking algorithms: the bootstrap particle filter, the unscented particle filter (both centralized and distributed schemes) and the distributed sigma-point information filter. They have been implemented and tested on simulated data, to evaluate the tracking performances for linear dynamic models and for both a Gaussian and a Rayleigh fading observation model. Extensive simulations showed that, overall, the target tracking system yields better accuracy for lower velocity of the target while performances get worse as the noise and the target velocity increase. Simulations also indicate that the algorithms here proposed compare favorably with previously proposed ones in terms of accuracy and computational complexity and energy consumption.

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless Sensor Networks: A Survey. Computer Networks (Elsevier), 393–422 (March 2002)
2. Ihler, A.T., et al.: Nonparametric belief propagation for self-localization of sensor networks. IEEE Journal on Selected Areas in Communications 23(4) (April 2005)
3. Lee, J., Cho, K., Lee, S., Kwon, T., Choi, Y.: Distributed and energy-efficient target localization and tracking in wireless sensor netwoks. Computer Communications 29, 2494–2505 (2006)
4. Vercauteren, T., Wang, X.: Decentralized Sigma-Point Information Filters for Target Tracking in Collaborative Sensor Networks. IEEE Transactions on Signal Processing 53(8) (August 2005)
5. Oshman, Y., Davidson, P.: Optimization of observer trajectories for bearings-only target localization. IEEE Transactions on Aerospace and Electronic Systems 35, 892–902 (1999)
6. Ertin, E., Fisher, J.W., Potter, L.C.: Maximum Mutual Information Principle for Dynamic Sensor Query Problems. In: Zhao, F., Guibas, L.J. (eds.) IPSN 2003. LNCS, vol. 2634, pp. 405–416. Springer, Heidelberg (2003)
7. Wang, H., Pottie, G., Yao, K., Estrin, D.: Entropy-based Sensor Selection Heuristic for Target Localization. In: International Workshop on Information Processing in Sensor Networks (IPSN), Berkeley, California, April 26-27 (2004)
8. Kaplan, L.M.: Global node selection for localization in a distributed sensor network. IEEE Transactions on Aerospace and Electronics Systems 42(1), 113–135 (2006)
9. Kaplan, L.M.: Local node selection for localization in a distributed sensor network. IEEE Transactions on Aerospace and Electronics Systems 42(1), 136–146 (2006)

10. Wang, Q., Chen, W.-P., Zheng, R., Lee, K., Sha, L.: Acoustic Target Tracking Using Tiny Wireless Sensor Devices. In: Zhao, F., Guibas, L.J. (eds.) IPSN 2003. LNCS, vol. 2634, pp. 642–657. Springer, Heidelberg (2003)
11. Chen, W.-P., Hou, J.C., Sha, L.: Dynamic Clustering for Acoustic Target Tracking in Wireless Sensor Networks. IEEE Trans. on Mobile Computing 3(3), 258–271 (2004)
12. Kung, H.T., Vlah, D.: Efficient Location Tracking Using Sensor Networks. In: WCNC (March 2003)
13. Lin, C.-Y., Tseng, Y.-C.: Structures for In-Network Moving Object Tracking in Wireless Sensor Networks. In: BROADNETS (2004)
14. Zhang, W., Cao, G.: DCTC: Dynamic Convoy Tree-Based Collaboration for Target Tracking in Sensor Networks. IEEE Trans. on Wireless Communications (2004)
15. Liu, J., Reich, J., Zhao, F.: Collaborative in-network processing for target tracking. EURASIP, J. Appl. Signal Processing, 378–391 (2003)
16. Zhao, F., Shin, J., Reich, J.: Information-driven dynamic sensor collaboration for tracking applications. IEEE Signal Processing Magazine 19(2), 61–72 (2002)
17. Kreucher, C.M., Hero, A.O., Kastella, K.D., Morelande, M.R.: An Information-Based Approach to sensor Management in Large Dynamic Networks. Proceeding of IEEE 95, 978–999 (2007)
18. Heinzelman, W.B., Chandrakasan, A.P., Balakrishnan, H.: Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In: Proceedings of 33rd Hawaii International Conference on System Sciences (HICSS 2000), Maui, Hawaii (January 2000)
19. Rappaport, T.S.: Wireless Communications. Principles and Practice, 2nd edn. Prentice Hall, Englewood Cliffs
20. Doucet, A., de Freitas, N., Gordon, N.: Sequential monte carlo methods in practice. Springer, Heidelberg (2001)
21. Gordon, N.J., Salmond, D.J., Smith, A.F.M.: Novel approach to nonlinear/nonlinear gaussian bayesian state estimation. IEEE Proceedings-F 140(2), 107–113 (1993)
22. Van der Merwe, R., Doucet, A., de Freitas, N., Wan, E.: The Unscented Particle Filter. In: NIPS, pp. 584–590 (2000)
23. Guo, D., Wang, X.: Dynamic sensor collaboration via sequential monte carlo. IEEE JSAC 22(6), 1037–1047 (2004)
24. Arienzo, L., Longo, M.: Energy-Efficient Tracking Strategy for Wireless Sensor Networks. In: Proceedings of IEEE MASS 2008, Workshop on Localized Communication and Topology Protocols for Ad hoc Networks, Atlanta, Georgia (September 2008)