

Renyi Entropy based Target Tracking in Mobile Sensor Networks

Dongbing Gu* Huosheng Hu*

* University of Essex, Wivenhoe Park, CO4 3SQ UK (e-mail:
dgu@essex.ac.uk, hhu@essex.ac.uk).

Abstract: This paper proposes an entropy based target tracking approach for mobile sensor networks. The proposed tracking algorithm runs a target state estimation stage and a motion control stage alternatively. A distributed particle filter is developed to estimate the target position in the first stage. This distributed particle filter does not require to transmit the weighted particles from one sensor node to another. Instead, a Gaussian mixture model is formulated to approximate the posterior distribution represented by the weighted particles via an *EM* algorithm. The *EM* algorithm is developed in a distributed form to compute the parameters of Gaussian mixture model via local communication, which leads to the distributed implementation of the particle filter. A flocking controller is developed to control the mobile sensor nodes to track the target in the second stage. The flocking control algorithm includes three components. Collision avoidance component is based on the design of a separation potential function. Alignment component is based on a consensus algorithm. Navigation component is based on the minimization of an quadratic Renyi entropy. The quadratic Renyi entropy of Gaussian mixture model has an analytical expression so that its optimization is feasible in mobile sensor networks. The proposed active tracking algorithm is tested in simulation.

Keywords: Target tracking, Particle filter, EM algorithm, Mobile sensor networks, Active control.

1. INTRODUCTION

Sensor networks provide a capability of detecting and estimating moving targets in the monitored environment-Martinerie (1997)Zhao et al. (2002). Particle filter is one of the widely used target estimation algorithms due to its applicability to non-linear and non-Gaussian dynamic systems Arulampalam et al. (2002)Doucet et al. (2001).

Currently there are several distributed particle filters (*DPFs*) that have been developed, see Coates (2004), Sheng et al. (2005), Zuo et al. (2006), Bashi et al. (2003). In these algorithms, the distributed nature is achieved by either transmitting local statistics of the particles to a centralized unit or using a message passing method. Transmitting local statistics of the particles to a centralized unit is not an efficient approach. It is also not robust. Failure of the centralized unit is vital to the entire network. In message passing method, the algorithms construct a path through the networks, which passes through all the nodes. Global statistics of the particles are accumulated by adding local statistics in each node through a forward pass. Then there needs a backward pass, which runs the important sampling and selection steps in each sensor node by using the accumulated global statistics. In Coates (2004), the factorized likelihood function was used and each partial likelihood function was updated at individual sensors using the local observations and the partial likelihood function estimated in the preceding sensors. The partial likelihood function was represented by a parameterized model and the parameters were transmitted through the path. The same strategy to communicate the

highly compact distribution was also used in robotics for map building in Rosencrantz et al. (2003). In Sheng et al. (2005), a set of uncorrelated sensor cliques was identified and they were automatically constructed according to moving target trajectories. The algorithm used a low dimensional Gaussian mixture model (*GMM*) to describe the posterior *pdf*. Model parameters rather than weighted particles were transmitted over the network. Using a *GMM* to approximate the posterior distribution was also adopted in Zuo et al. (2006) where the estimated parameters of *GMM* were transmitted to a fusion center. In Bashi et al. (2003), the particles were distributed in a sensor network, i.e. each sensor node held part of the particles. Local statistics of the particles were calculated and transmitted to a centralized unit. In our previous work Gu (2007), a *DPF* was developed by first using a *GMM* to approximate the posterior distribution and then using an averaging consensus filter to estimate the parameters of *GMM* via an *EM* algorithm.

Given the result from target estimation, the mobile sensor networks make collective motion decisions to track the target. Based on our developed *DPF*, this paper proposes a tracking control algorithm for mobile sensor networks. The tracking control signal is obtained by optimizing an information measure to reduce the uncertainty of posterior probability. Using information measure to guide multiple mobile agents for active sensing have been researched in Grocholsky et al. (2003), Chung et al. (2004) where the Shannon entropy was directly obtained from the determinant of posterior Gaussian covariance of Kalman filter. However, the Shannon entropy can not be computed di-

rectly in particle filter. Recently the Shannon entropy was derived in Ryan and Hedrick (2010) for a single mobile agent by using a piecewise linear approximation to the posterior *pdf* in particle filter. In Hoffmann and Tomlin (2010), a mutual information gain was used as the information measure for multiple mobile agents. The computation of mutual information was implemented by integrating the likelihood function weighted by particle weights over entire measurement range.

The information entropy can capture the higher order moments of a probability density function, rather than operating on the second order statistics. On this aspect, the Renyi entropy plays the same role as the Shannon entropy does. When a *GMM* distribution is concerned, applying the Shannon entropy as a criterion will result in a difficult computational problem, while the Renyi entropy has an analytical expression. In this paper, we propose to use a quadratic Renyi entropy as the information measure to guide the motion of mobile sensor networks. As the posterior *GMM* in our *PDF* is already computed in distributed way, the optimization of the Renyi entropy is feasible. Building on the result of optimizing the Renyi entropy, a flocking algorithm is developed, which include two additional control components: collision avoidance and motion alignment.

In the rest of this paper, the particle filter and *EM* algorithm in a sensor network setting are described in Section 2. The distributed particle filter is given in Section 3. The Renyi entropy based tracking control is presented in Section 4. Section 5 provides simulation results. Finally, conclusions are summarized in Section 6.

2. PARTICLE FILTER AND *EM* ALGORITHM

2.1 Particle Filter in Sensor Networks

We consider a network of M sensors, which is used to track a moving target. The moving target is modeled by a discrete state equation:

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{v}_{k-1} \quad (1)$$

where \mathbf{x}_k is the n_x dimensional state vector and \mathbf{v}_k is the Gaussian noise with mean zero and covariance Q_k . The state can also be modeled as a Markov process with an initial distribution $p(\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0, Q_0)$ and a state transition probability $p(\mathbf{x}_k|\mathbf{x}_{k-1})$.

Each sensor i can make n_z dimensional measurement $\mathbf{y}_{i,k}$ ($i = 1, \dots, M$) at time k . The measurement state equation is assumed as follows:

$$\mathbf{y}_{i,k} = \mathbf{h}_i(\mathbf{x}_k) + \mathbf{w}_k \quad (2)$$

where \mathbf{w}_k is the Gaussian noise with mean zero and covariance R_k . The measurement equation (2) can also be modeled as a likelihood function $p(\mathbf{y}_{i,k}|\mathbf{x}_k)$. It is assumed that the state noise and measurement noise are independent, $E[\mathbf{v}_k \mathbf{w}_k^T] = 0$.

Let $\mathbf{x}_{0:k}$ denote $\{\mathbf{x}_t, t = 0, \dots, k\}$ and $\mathbf{y}_{i,1:k}$ denote $\{\mathbf{y}_{i,t}, t = 0, \dots, k\}$. The particle filter is to estimate the posterior *pdf* $p(\mathbf{x}_k|\mathbf{y}_{i,1:k})$. Now we assume that a centralized unit exists and it receives one measurement from each sensor node at each k . The centralized unit maintains a set of weighted particles $\{\mathbf{x}_k^{(n)}, \omega_k^{(n)}\}$. When the centralized

unit receives the measurements $\mathbf{y}_{i,k}$ from $i = 1, \dots, M$ at k , it predicts the particles $\mathbf{x}_k^{(n)}$ and update the weights $\omega_k^{(n)}$, ($n = 1, \dots, N$), where N is the number of particles.

Let $\{\mathbf{x}_k^{(n)}, \omega_k^{(n)}\}_{n=1}^N$ denote the posterior *pdf* $p(\mathbf{x}_k|\mathbf{y}_{i,1:k})$, where the weights are normalized, $\sum_{n=1}^N \omega_k^{(n)} = 1$. The posterior *pdf* can be approximated expressed as:

$$p(\mathbf{x}_k|\mathbf{y}_{i,1:k}) \approx \omega_k^{(n)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(n)}) \quad (3)$$

where δ denotes the Dirac delta function. There are two steps in a particle filter. The first is the prediction step where all the particles move to their next step positions via the state transition probability:

$$\mathbf{x}_k^{(n)} \sim p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(n)}) \quad (4)$$

The second is the updating step where the measurements are made and the weights are updated according to the likelihood function:

$$\omega_k^{(n)} \propto \omega_{k-1}^{(n)} p(\mathbf{y}_{i,k}|\mathbf{x}_k^{(n)}) \quad (5)$$

2.2 *EM* algorithm for Gaussian mixtures

From now on, the centralized unit discussed above is removed. It is assumed that each sensor maintains N weighted particles $\{\mathbf{x}_{i,k}^{(n)}, \omega_{i,k}^{(n)}\}$. In total, there are NM particles in the network. The posterior *pdf* of the particle filter is represented with NM weighted particles. When more sensor nodes are available in the network, there will be more weighted particles to represent the posterior *pdf* and therefore resulting in a more accurate estimate. However, it is not feasible to transmit all the weighted particles over the entire network due to the limitation of wireless communication. In this paper, a *GMM* is to be learned from all the weighted particles via a distributed *EM* algorithm. In the distributed *EM* algorithm, only some local statistics of the weighted particles are required to exchange between neighbor nodes. We present an *EM* algorithm to formulate the *GMM* from all the weighted particles in this subsection. The distributed *EM* will be detailed in next section.

A *GMM* with C mixture probabilities $\alpha_{k,c}$, ($c = 1, \dots, C$) can be found from the weighted particles via *EM* algorithm at step k . It will be shown later that $\alpha_{k,c} = \sum_{i=1}^M N \alpha_{i,k,c}$. The *GMM* for particle $\mathbf{x}_{i,k}$ is

$$p(\mathbf{x}_{i,k}|\theta_k) = \sum_{c=1}^C \alpha_{i,k,c} p(\mathbf{x}_{i,k}|\mu_{k,c}, \Sigma_{k,c}) \quad (6)$$

where θ_k is a parameter set of *GMM* to be estimated, $\theta_k = \{\alpha_{k,c}, \mu_{k,c}, \Sigma_{k,c}; c = 1, \dots, C\}$, and

$$\begin{aligned} p(\mathbf{x}_{i,k}|\mu_{k,c}, \Sigma_{k,c}) \\ = \frac{1}{(2\pi)^{\frac{n_x}{2}} |\Sigma_{k,c}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}_{i,k} - \mu_{k,c})^T \Sigma_{k,c}^{-1} (\mathbf{x}_{i,k} - \mu_{k,c})} \end{aligned}$$

The *EM* algorithm updates θ_k between two consecutive sensor observations at k and $k+1$ of the particle filter. Subscript t is used inside the loop of the *EM* algorithm to indicate the time step starting from k and ending with $k+1$.

In the E step, the mixture probabilities are updated:

$$\alpha_{i,t+1,c}^{(n)} = \frac{\alpha_{i,t,c}^{(n)} \mathcal{P}(\mathbf{x}_{i,k}^{(n)} | \mu_{t,c}, \Sigma_{t,c})}{\sum_{c=1}^C \alpha_{i,t,c}^{(n)} \mathcal{P}(\mathbf{x}_{i,k}^{(n)} | \mu_{t,c}, \Sigma_{t,c})} \quad (7)$$

In the M step, the parameter set is updated:

$$\begin{aligned} \alpha_{i,t+1,c} &= \frac{1}{N} \sum_{n=1}^N \alpha_{i,t+1,c}^{(n)} \\ \mu_{t+1,c} &= \frac{\sum_{i=1}^M \sum_{n=1}^N \alpha_{i,t+1,c}^{(n)} \mathbf{x}_{i,k}^{(n)}}{\sum_{i=1}^M \sum_{n=1}^N \alpha_{i,t+1,c}^{(n)}} \\ \Sigma_{t+1,c} &= \frac{\sum_{i=1}^M \sum_{n=1}^N \alpha_{i,t+1,c}^{(n)} (\mathbf{x}_{i,k}^{(n)} - \mu_{t+1,c})(\mathbf{x}_{i,k}^{(n)} - \mu_{t+1,c})^T}{\sum_{i=1}^M \sum_{n=1}^N \alpha_{i,t+1,c}^{(n)}} \end{aligned} \quad (8)$$

The above iteration equations can be further written as a compact form:

$$\mu_{t+1,c} = \frac{\sum_{i=1}^M a_{i,t+1,c}}{\sum_{i=1}^M N \alpha_{i,t+1,c}} \quad \Sigma_{t+1,c} = \frac{\sum_{i=1}^M b_{i,t+1,c}}{\sum_{i=1}^M N \alpha_{i,t+1,c}} \quad (9)$$

where the local statistics is defined as:

$$\begin{aligned} \alpha_{i,t,c} &= \frac{1}{N} \sum_{n=1}^N \alpha_{i,t,c}^{(n)} \\ a_{i,t,c} &= \sum_{n=1}^N \alpha_{i,t,c}^{(n)} \mathbf{x}_{i,k}^{(n)} \\ b_{i,t,c} &= \sum_{n=1}^N \alpha_{i,t,c}^{(n)} (\mathbf{x}_{i,k}^{(n)} - \mu_{t,c})(\mathbf{x}_{i,k}^{(n)} - \mu_{t,c})^T \end{aligned} \quad (10)$$

Accordingly, the global statistics is defined as:

$$\begin{aligned} \alpha_{t,c} &= \sum_{i=1}^M N \alpha_{i,t,c} \\ a_{t,c} &= \sum_{i=1}^M a_{i,t,c} \\ b_{t,c} &= \sum_{i=1}^M b_{i,t,c} \end{aligned} \quad (11)$$

Using the global quantities defined above, the estimated parameters are:

$$\mu_{t+1,c} = \frac{a_{t+1,c}}{\alpha_{t+1,c}} \quad \Sigma_{t+1,c} = \frac{b_{t+1,c}}{\alpha_{t+1,c}} \quad (12)$$

3. DISTRIBUTED PARTICLE FILTER

A mobile sensor network can be modeled by using algebraic graph theory. A graph can be used to represent the interconnections between sensor nodes. A vertex of the graph corresponds to a node and edges of the graph capture the dependence of interconnections. Formally, a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of vertices $\mathcal{V} = \{1, \dots, M\}$ and a set of edges $\mathcal{E} = \{e(i, j) \in \mathcal{V} \times \mathcal{V}\}$, containing unordered pairs of distinct vertices. Assuming the graph has no loops, i.e. $e(i, j) \in \mathcal{E}$ implies $i \neq j$.

Let D denote the distance that a node can communicate via wireless radio links. Edge $e(i, j)$ is connected if the Euclidean distance $r_{k,ij}$ between nodes i and j is less than or equal to D .

A graph is connected if for any vertices $e(i, j) \in \mathcal{V}$, there exists a path of edges in \mathcal{E} from i to j . The set of neighbors of vertex i is defined as $\mathcal{N}_i = \{j \in \mathcal{V} : e(i, j) \in \mathcal{E}\}$. The degree of vertex i is defined as $|\mathcal{N}_i|$ and maximum degree is $d_{max} = \max_i(|\mathcal{N}_i|)$. Let Δ be the degree matrix, $\Delta = \text{diag}(|\mathcal{N}_i|)$. The adjacency matrix \mathcal{A} is the integer matrix with rows and columns indexed by the vertices, such as the ij -entry of \mathcal{A} is equal to the number of edges from i to j . Following Godsil and Royle (2001), Laplacian matrix of a graph \mathcal{G} is defined as L :

$$L = \Delta - \mathcal{A}$$

It should be noted that L varies with time step k , i.e. it varies with the motion of sensor nodes. But for the EM algorithm, it is kept constant as the EM run between k and $k + 1$. For a connected graph, Laplacian matrix L is symmetric and positive semi-definite. Its minimum eigenvalue is 0 and the corresponding eigenvector is $\mathbf{1} = [1, \dots, 1]^T$ or $L\mathbf{1} = 0$, see Godsil and Royle (2001).

In the EM algorithm mentioned above, it can be found that the local statistics can be calculated locally, while the global statistics can not be calculated locally. However, the global statistics can be viewed as the averages of the local statistics from all the nodes in equation (10). This view can be made more clear by redefining the global statistics in (11) as the average as follows:

$$\begin{aligned} \alpha_{t,c} &= \frac{1}{M} \sum_{i=1}^M N \alpha_{i,t,c} \\ a_{t,c} &= \frac{1}{M} \sum_{i=1}^M a_{i,t,c} \\ b_{t,c} &= \frac{1}{M} \sum_{i=1}^M b_{i,t,c} \end{aligned} \quad (13)$$

This redefinition does not affect the parameter estimation in equation (12).

Due to the average expressions in (13), the idea of the average consensus filter proposed in Lynch (1996), Olfati-Saber and Murray (2004) can be used to estimate the global statistics through information diffusion over the network. Each node exchanges the local statistics with its neighbors and estimates the global statistics based on neighbor's local statistics through the consensus filter.

Let $\{\bar{\alpha}_{i,t,c}, \bar{a}_{i,t,c}, \bar{b}_{i,t,c}\}$ denote the estimates of the global statistics $\{\alpha_{t,c}, a_{t,c}, b_{t,c}\}$ in node i . Let vector $\zeta_{i,t,c}$ denote one of the estimates of the global statistics, $\bar{\alpha}_{i,t,c}$, $\bar{a}_{i,t,c}$, or $\bar{b}_{i,t,c}$. Let vector $\mathbf{g}_{i,t,c}$ denote one of the local statistics $N\alpha_{i,t,c}$, $a_{i,t,c}$, or $b_{i,t,c}$. The consensus filter in node i takes as inputs the local statistics $\mathbf{g}_{i,t,c}$. It outputs the estimated global statistics $\zeta_{i,t,c}$.

An average consensus filter in a sensor node i is designed as follows in the discrete form:

$$\zeta_{i,t+1,c} = \zeta_{i,t,c} + \epsilon \left[\sum_{j \in \mathcal{N}_i} (\zeta_{j,t,c} - \zeta_{i,t,c}) + (\mathbf{g}_{i,t,c} - \zeta_{i,t,c}) \right] \quad (14)$$

where ϵ is the updating rate and should be

$$\epsilon \leq \frac{1}{d_{max}}$$

This requirement guarantees the stability of the discrete consensus filter according to the Gersgorin theorem. $\zeta_{i,t,c}$ can asymptotically converge to the average of the local inputs $\mathbf{g}_{i,t,c}$:

$$\zeta_{i,t,c} \rightarrow \frac{1}{M} \sum_{i=1}^M \mathbf{g}_{i,t,c} \quad (15)$$

Since $\zeta_{i,t,c}$ represents the estimates of the global statistics and $\mathbf{g}_{i,t,c}$ represents the local statistics, we have:

$$\begin{aligned} \bar{\alpha}_{i,t,c} &\rightarrow \alpha_{i,t,c} = \sum_{i=1}^M N \alpha_{i,t,c} \\ \bar{a}_{i,t,c} &\rightarrow a_{t,c} = \sum_{i=1}^M a_{i,t,c} \\ \bar{b}_{i,t,c} &\rightarrow b_{t,c} = \sum_{i=1}^M b_{i,t,c} \end{aligned} \quad (16)$$

The estimated parameters are:

$$\bar{\mu}_{t+1,c} = \frac{\bar{\alpha}_{i,t,c}}{\bar{\alpha}_{i,t,c}} \quad \bar{\Sigma}_{t+1,c} = \frac{\bar{b}_{i,t,c}}{\bar{\alpha}_{i,t,c}} \quad (17)$$

Once the estimated global statistics are obtained, the particles can be drawn from the *GMM*. Then they should be propagated through the state transition probability to generate the predicted particles. With the predicted particles, the weight updating step and re-sampling step are the same as the steps presented in Section 2.

4. ACTIVE TRACKING CONTROL

4.1 Flocking Controller

The mobility of sensor nodes is described by double integrator dynamics. For node i with $2D$ dimensional coordinates $q_{i,k}$ and speed vector $p_{i,k}$, the state and control vectors are $\mathbf{z}_{i,k} = [q_{i,k}, p_{i,k}]^T$ and $\mathbf{u}_{i,k}$. The double integrator dynamics is:

$$\mathbf{z}_{i,k+1} = A\mathbf{z}_{i,k} + B\mathbf{u}_{i,k} \quad (18)$$

where

$$A = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ T & 0 \\ 0 & T \end{bmatrix}$$

and T is the sample interval.

As a group of mobile nodes, it is necessary to have a flocking behavior, which can avoid the potential collisions between group members, align their headings with the group motion direction, and navigate towards the best positions to observe the target.

The collision avoidance control is designed by using a separation potential function. The requirement of this control is to keep adjacent mobile nodes with a specific distance d_0 . If the distance $r_{k,ij}$ between adjacent nodes is smaller than d_0 , they attempt to separate. Let H_s denote the separation potential function between nodes i and j :

$$H_s = \sum_{i=1}^M \sum_{j \neq i} \frac{1}{2} \left(\log(\kappa + r_{k,ij}) + \frac{\kappa + d^2}{\kappa + r_{k,ij}} \right) \quad (19)$$

where $\kappa > 0$. This potential function has a minimum point where $r_{k,ij} = d$. d is the desired separation distance and smaller than, but close to d_0 . When $r_{k,ij}$ is larger than d_0 , no contribution from this potential function to the control signal. The gradient of the separation potential function with respect to $q_{i,k}$ is:

$$\frac{\partial H_s}{\partial q_{i,k}} = \begin{cases} \frac{1}{2} \sum_{j \in \mathcal{N}_i} \frac{(r_{k,ij} - d^2)(q_i - q_j)}{\kappa + r_{k,ij}^2} & \text{if } r_{k,ij} < d_0 \\ 0 & \text{otherwise} \end{cases} \quad (20)$$

The alignment control is designed by controlling the individual heading to align with neighbor's headings. This is implemented by minimizing a speed disagreement function:

$$H_c = \frac{1}{2} \sum_{i,j \in \mathcal{E}} \|p_{i,k} - p_{j,k}\|^2 \quad (21)$$

The gradient of the disagreement function with respect to $q_{i,k}$ is:

$$\frac{\partial H_c}{\partial p_{i,k}} = \frac{1}{2} \sum_{j \in \mathcal{N}_i} (p_{i,k} - p_{j,k}) \quad (22)$$

More details on these two control components can be found from Olfati-Saber and Murray (2004), Tanner et al. (2007). The navigation control is designed according to the optimization of an information measure discussed below.

4.2 Renyi Entropy

The Renyi entropy is a generalization of the Shannon entropy.

$$H_r = \frac{1}{1-r} \log \int p(\mathbf{x})^r d\mathbf{x}$$

When r approaches to 1, H_1 converges to the Shannon entropy. When $r = 2$, the quadratic Renyi entropy H_2 is defined as:

$$H_2 = -\log \int p(\mathbf{x})^2 d\mathbf{x} \quad (23)$$

Given the *GMM* of posterior *pdf* $p(\mathbf{x}_k | \mu_{k,c}, \Sigma_{k,c})$ generated in *DPF*, the quadratic Renyi entropy has the following analytical expression:

$$\begin{aligned} H_2 &= -\log \int \left[\sum_{c=1}^C \alpha_{k,c} p(\mathbf{x}_k | \mu_{k,c}, \Sigma_{k,c}) \right]^2 d\mathbf{x}_k \\ &= -\log \left[\sum_{i=1}^C \sum_{j=1}^C \alpha_{k,i}^T K_{k,ij} \alpha_{k,j} \right] \\ &= -\log[\alpha_k^T K_k \alpha_k] \end{aligned} \quad (24)$$

where K_k is a symmetric matrix of elements $K_{k,ij} = \mathcal{N}(\mu_i - \mu_j, P_i + P_j)$.

Actually, the measurement equation (2) depends on both the target position \mathbf{x}_k and the mobile node's position $\mathbf{q}_{i,k}$:

$$\mathbf{y}_{i,k} = \mathbf{h}_i(\mathbf{x}_k, \mathbf{q}_{i,k}) + \mathbf{w}_k$$

That is the likelihood probability $p(\mathbf{y}_{i,k}|\mathbf{x}_k, \mathbf{q}_{i,k})$ depends on the mobile node's position $\mathbf{q}_{i,k}$ as well. Accordingly, the *GMM* parameter θ_k depends on $\mathbf{q}_{i,k}$ as it is obtained from the *EM* algorithm, which updates the particle weights by using the likelihood probability $p(\mathbf{y}_{i,k}|\mathbf{x}_k, \mathbf{q}_{i,k})$. So α_k and K_k in H_2 depend on the mobile node's position $\mathbf{q}_{i,k}$. Thus a navigation control can be designed by optimizing the following problem:

$$\mathbf{q}_{i,k} = \arg \min H_2$$

The gradient of H_2 with respect to $\mathbf{q}_{i,k}$ can be computed both analytically and numerically. In our simulation, it is computed numerically. The maximum distance traversable by the mobile node in a single time step is used to find a desired position where H_2 reaches its minimum value. The mobile node is controlled to move in the direction towards the desired position.

In summary, the control input signal to the mobile sensor node i is:

$$\mathbf{u}_{i,k} = -k_1 \frac{\partial H_s}{\partial \mathbf{q}_{i,k}} - k_2 \frac{\partial H_c}{\partial \mathbf{p}_{i,k}} - k_3 \frac{\partial H_2}{\partial \mathbf{q}_{i,k}}$$

where k_1, k_2, k_3 are positive gains.

5. SIMULATIONS

A mobile sensor network with 10 nodes ($M = 10$) was used for simulation. They were randomly placed in a square area $[0, 1m] \times [0, 1m]$. The communication distance was set as $D = 0.8m$. $N = 50$ weighted particles were used in each node. The posterior distribution *GMM* is assumed to have $C = 3$ mixture probabilities. The parameter estimation of the distributed *EM* algorithm was executed 10 loops between two consecutive observations. T was selected as 0.05s. The parameters of separation potential function were set as $d_0 = 0.3, d = 0.25, \kappa = 0.05$. A linear measurement equation is used:

$$\mathbf{y}_{i,k} = \mathbf{h}\mathbf{x}_k + \mathbf{w}_k$$

where $\mathbf{h} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. The measurement covariance R_k is modeled as a non-linear function of distance $\rho_{i,k}$:

$$R_k(\rho_{i,k}) = 0.015 \left(11 + 9 \frac{(\rho_{i,k} - \gamma)}{\sqrt{1 + (\rho_{i,k} - \gamma)^2}} \right)$$

where $\rho_{i,k} = \|\mathbf{x}_k - \mathbf{q}_{i,k}\|$ is the Euclidean distance between the target and node i . This function is to make the measurement noise increase with the increase of distance.

A moving target with the following state equation was used in simulation:

$$\mathbf{x}_k = A\mathbf{x}_{k-1} + B\mathbf{v}_{k-1}$$

where A is the same as in (18) and $B = \begin{bmatrix} T^2/2 & 0 \\ 0 & T^2/2 \\ T & 0 \\ 0 & T \end{bmatrix}$.

and the covariances of Gaussian noises \mathbf{v}_k was selected as:

$$Q_k = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix}.$$

The simulation was run for 500 loops. The target started from (5,0) and the true and estimated target trajectories in one of node are shown Fig. 1. It can be seen that the estimated trajectory (blue line) was very close to the true trajectory (black line) although the estimated trajectory was not very smooth. The estimated particles at the beginning and the end of tracking are also shown in Fig. 1 (see the black dot clusters). They represented the estimated covariance changes from a large cluster at the beginning to a small cluster at the end.

The true target trajectory and 10 mobile node trajectories are shown in Fig. 2. As the mobile nodes were initially placed far away from the target, they were controlled to move close to the target very quickly at the beginning. Then they were able to follow the target without lost. Also there were no collision between neighbors during the entire tracking process. The distance between the target and the center of flocking nodes is shown Fig. 3, which clearly demonstrated the tracking behavior observed in Fig. 2. The quadratic Renyi entropy was also calculated in each loop from one of the nodes and is shown in Fig. 4. The decrease before around 100 loops shows the mobile nodes had less information and they increased their speeds to gain more information. After around 100 loops, the quadratic Renyi entropy varied around a certain value, demonstrating a behavior of tracking the target. The variations show the stochastic nature of the entropy.

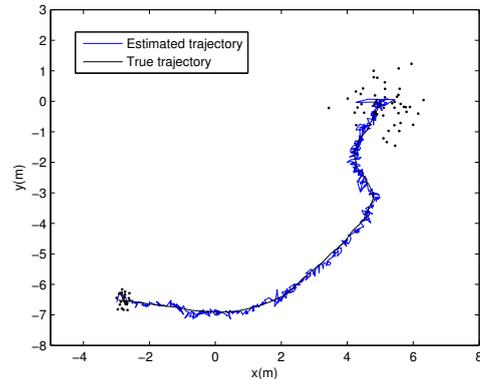


Fig. 1. True and Estimated Trajectories

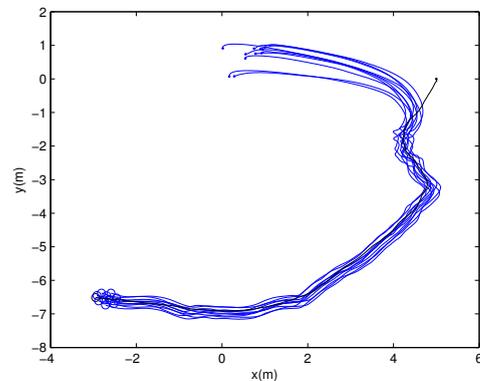


Fig. 2. True target trajectory and 10 mobile node trajectories

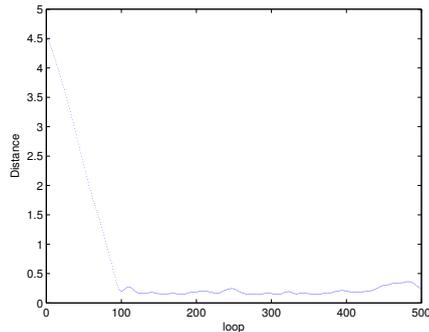


Fig. 3. Distance between target and the center of sensor nodes

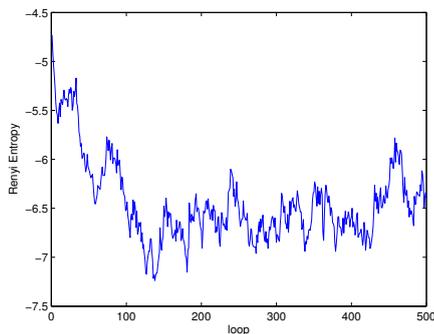


Fig. 4. Quadratic Renyi entropy

6. CONCLUSION

This paper investigates a particle filter approach to estimating a target and an entropy based active control approach to tracking the estimated target. Several techniques are employed in order to implement them in a mobile sensor network. Firstly the *GMM* of the particle filter is computed via an distributed *EM* algorithm so that a *PDF* is obtained. Secondly a quadratic Renyi entropy is used as a information measure to guide the navigation of mobile sensor nodes. This is possible because the analytical expression of the quadratic Renyi entropy for *GMM* exists. Also the entropy based active navigation control is combined with collision avoidance behavior and motion alignment behavior to implement a flocking control for the mobile sensor nodes.

Currently one step horizon is used in optimizing the quadratic Renyi entropy. In our next work, we would like to explore the possibility of using multiple step horizon for optimizing the quadratic Renyi entropy so that a long term prediction is obtained to improve the performance.

ACKNOWLEDGEMENTS

This research work is financially sponsored by European Union FP7 program, ICT-231646, SHOAL.

REFERENCES

Arulampalam, M., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Processing*, 50, 174–188.

- Bashi, A.S., Jilkov, V.P., Li, X.R., and Chen, H. (2003). Distributed implementations of particle filters. In *Proc. 2003 International Conf. Information Fusion*, 1164–1171. Cairns, Australia.
- Chung, T.H., Gupta, V., Burdick, J.W., and Murray, R.M. (2004). On a decentralized active sensing strategy using mobile sensor platforms in a network. In *Proc. 43rd IEEE Conf. Decision Control*, 1914–1919. Pasadena, CA.
- Coates, M.J. (2004). Distributed particle filtering for sensor networks. In *Proc. of Int. Symp. Information Processing in Sensor Networks (IPSN)*. Berkeley, CA.
- Doucet, A., de Freitas, N., and Gordon, N. (2001). *Sequential Monte Carlo methods*. Springer-Verlag, New York.
- Godsil, C. and Royle, G. (2001). *Algebraic graph theory*. Springer-Verlag.
- Grocholsky, B., Makarenko, A., and Durrant-Whyte, H. (2003). Information-theoretic coordinated control of multiple sensor platforms. In *Proc. IEEE Int. Conf. Robotics and Automation*, 1521–1526. Taipei, Taiwan.
- Gu, D. (2007). Distributed particle filter for target tracking. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*. Rome, Italy.
- Hoffmann, G.M. and Tomlin, C. (2010). Mobile sensor network control using mutual information methods and particle filters. *IEEE Trans. on Automatic Control*, 55, 32–47.
- Lynch, N.A. (1996). *Distributed algorithms*. Morgan Kaufmann Publishers, Inc., San Francisco, CA.
- Martinerie, F. (1997). Data fusion and tracking using HMMs in a distributed sensor network. *IEEE Trans. Aerospace and Electronic Systems*, 33, 11–28.
- Olfati-Saber, R. and Murray, R.M. (2004). Consensus problems in networks of agents with switching topology and time-delay. *IEEE Trans. on Automatic Control*, 49(9), 101–115.
- Rosencrantz, M., Gordon, G., and Thrun, S. (2003). Decentralized sensor fusion with distributed particle filters. In *Proc. of Conf. Uncertainty in Artificial Intelligence*. Acapulco, Mexico.
- Ryan, A. and Hedrick, J.K. (2010). Particle filter based information-theoretic active sensing. *Robotics and Autonomous Systems*, 58(5), 574–584.
- Sheng, Y., Hu, X., and Ramanathan, P. (2005). Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor networks. In *Proc. of the 4th Int. Symposium on Information Processing in Sensor Networks*.
- Tanner, H.G., Jadbabaie, A., and Pappas, G.J. (2007). Flocking in fixed and switching networks. *IEEE Trans. Automatic Control*, 52(5), 863–868.
- Zhao, F., Shin, J., and Reich, J. (2002). Information-driven dynamic sensor collaboration for tracking applications. *IEEE Signal Processing Magazine*, 19, 61–72.
- Zuo, L., Mehrotra, K., Varshney, P., and Mohan, C. (2006). Bandwidth-efficient target tracking in distributed sensor networks using particle filters. In *Proc. of 14th European Signal Processing Conference EURASIP2006*. Florence, Italy.