# Mobility Assisted Distributed Tracking in Hybrid Sensor Networks

Thakshila Wimalajeewa, *Student Member, IEEE* and Sudharman K. Jayaweera, *Senior Member, IEEE*

Department of Electrical and Computer Engineering,
University of New Mexico, NM 87131.
Email: {thakshilaw, jayaweera}@ece.unm.edu

***Abstract*—In this paper, we propose a new mobility assisted tracking (MAT) algorithm for tracking a single target in a hybrid sensor network consisting of both static and mobile nodes. The network is assumed to be partitioned into clusters and cluster heads are formed from a set of high capacity static nodes. One cluster head is selected to perform the tracking task using particle filters at a given time based on the observations received from the nodes belonging to the corresponding cluster. We exploit the node mobility in the hybrid sensor network to dynamically maintain a certain *coverage level* at the predicted target location at each time. In the proposed MAT algorithm mobile nodes are directed to move towards the predicted target position at each time step if the predicted target position is not covered to the desired *coverage level* by static nodes. Simulation results show that with the proposed MAT algorithm, an improved performance closer to the PCRLB is achieved with a relatively small number of mobile nodes in the network compared to the scenario when all nodes are static. The proposed scheme is also robust against static node as well as cluster head failures.**

## I. INTRODUCTION

Use of particle filtering for target tracking was addressed by many authors in recent research with static sensor networks [1]–[3]. For example, in [2] target tracking based on binary observations in a static sensor network was considered, where the tracking is performed at a central fusion center. In [3], tracking algorithm based on particle filters incorporating imperfect communication between sensor nodes and the fusion center is proposed. When the target tracking is performed at a centralized fusion center, each node in the network should forward its raw or locally processed observations to the central unit, perhaps via long-range communication, which indeed consumes a large transmit power at nodes. Since many practical sensor networks are operated with sensor nodes which have limited battery power, it is desirable that the tracking is performed distributively utilizing the limited resources efficiently. Distributed implementation of particle filters is proposed in some recent work [4], [5]. In those distributed particle filter approaches, all nodes in the network are active at every time and participate in the tracking task. However, all the nodes in the network might not have rich information regarding the target state as the target moves. Thus it is of interest to obtain observations from sensors which have useful information allowing the rest of the nodes to be idle saving energy.

On the other hand, most of the existing work on target tracking consider only static sensor networks. However, stationary sensor networks may not suit for some applications. For example, in situations where it might be necessary to deploy a large number of static nodes to monitor a large region with a desired performance level. Target tracking in mobile sensor networks is addressed recently in [6]–[9]. In [6], [7], the tracking task was performed based on Kalman filters assuming linear dynamic models and information driven approaches for mobility management are presented. Target tracking with particle filters in a mobile sensor network based on a centralized approach was considered in [8], [9].

In this paper, our focus is on developing a target tracking algorithm based on distributed particle filtering in a hybrid sensor network consisting of both static and mobile nodes, and use of node mobility to compensate for the lack of coverage provided by static nodes dynamically as the target moves. To the best of our knowledge, a tracking algorithm in a hybrid sensor network exploiting the node mobility dynamically to compensate for the lack of coverage by static nodes, is not addressed in the literature. In the proposed mobility assisted tracking (MAT) scheme, the network is partitioned into clusters and only one cluster head is active at a given time. Since nodes have to communicate with only their cluster heads, they do not necessarily need to have long communication ranges. The active cluster head is selected based on the predicted target locations. When the active cluster head is selected, the associate sensor nodes are activated and asked to send their local measurements to the corresponding cluster head. In the proposed MAT algorithm, the main idea is to maintain a certain *coverage level* for the predicted target location at each time step. By *coverage level*, we mean that each predicted target location at time $k$, is covered by exactly or approximately by a certain number (say $\beta$) of sensor nodes. The predicted target location is $\beta$ covered essentially means that there is at least $\beta$ number of nodes located within a certain distance (which is a design parameter and is discussed in Section IV in detail) from the predicted target location. The terms $exact$ and $approximate$ coverage are explained in a later section of the paper. If the predicted position at time $k$ is already covered by $\beta$-number of static nodes, then mobile nodes are not needed to move during time $k$ to $k+1$. Otherwise, to maintain the $\beta$-coverage (exact or approximate) mobile nodes are directed to move taking the energy and speed constraints into account.

The paper is organized as follows. Section II presents the system model and problem formulation. In Section III, the cluster based distributed target tracking by particle filtering is explained. Proposed node mobility management scheme is discussed in Section IV. Performance results are shown in V and concluding remarks are given in VI

[1]This research was supported in part by the National Science foundation (NSF) under the grant CCF-0830545.

## II. Problem Formulation and System Model

### A. Sensor network model

We consider a hybrid sensor network consisting of $n_s$ number of static nodes and $n_m$ number of mobile nodes. Denote $n = n_s + n_m$ to be the total number of nodes. In general we assume that $n_m << n_s$, since deploying a large number of mobile nodes is not as cost effective as deploying static nodes. Let $\mathcal{V}$ be the set containing all node indices in the network and $\mathcal{V}_m$ and $\mathcal{V}_s$ be the sets containing mobile and static node indices, respectively.

### B. State dynamics model

We consider the problem of tracking a single target which is moving in 2-dimensional $X \times Y$ plane. Denote $\mathbf{x}_k = [x_{1k}\ x_{2k}\ \dot{x_{1k}}\ \dot{x_{2k}}]^T$ to be the target state vector at time $k$ where first two elements represent the target position and the latter two elements of $\mathbf{x}_k$ represent the speed of the target in $X$ and $Y$ directions, respectively. We assume following discrete time linear dynamical model for the target state:

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{u}_k, \tag{1}$$

for $k = 1, 2, \cdots$, with the initial known distribution, $p(\mathbf{x}_0)$ for $\mathbf{x}_0$ where $\mathbf{F}_k$ is a $4 \times 4$ matrix that models the state kinematics and is defined as, [10]

$$\mathbf{F}_k = \begin{pmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where $T_s$ is the time difference between two consecutive measurements (or sampling period). The noise vector $\mathbf{u}_k$ is assumed to be zero mean Gaussian with covariance matrix $Q$ where $Q$ is given by [10],

$$Q = \sigma_u^2 \begin{pmatrix} \frac{T_s^3}{3} & 0 & \frac{T_s^2}{2} & 0 \\ 0 & \frac{T_s^3}{3} & 0 & \frac{T_s^2}{2} \\ \frac{T_s^2}{2} & 0 & T_s & 0 \\ 0 & \frac{T_s^2}{2} & 0 & T_s \end{pmatrix},$$

which models the acceleration terms in $X$ and $Y$ directions. $\sigma_u^2$ is a scalar which controls the intensity of the process noise. We assume that the dynamic model (1) performs a Markov transition and is represented by the conditional transition probability density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$.

### C. Observation model

We assume that the signal emitted by the target is attenuated with the distance from the target according to the following model:

$$z_{j,k} \;=\; \frac{A_0}{|\mathbf{r}_{j,k} - \tilde{\mathbf{x}}_k|^{\alpha/2}} + \nu_{j,k},\ \ for\ j \in \mathcal{V} \tag{2}$$

where $z_{j,k}$ is the $j$-th node's observation at time $k$, $\tilde{\mathbf{x}}_k = [\mathbf{x}_k(1)\ \ \mathbf{x}_k(2)]^T = [x_{1k}\ \ x_{2k}]^T$, $\mathbf{r}_{j,k} = (r_{j,k}, s_{j,k})$ is the position of the $j$-th node at time $k$, $A_0$ is the amplitude of the signal emitted by the target and $\nu_{j,k}$ is the observation noise which is assumed to be zero mean Gaussian with variance $\sigma_\nu^2$ and independent across sensor nodes, and $\alpha$ is the path-loss attenuation index, which is assumed to be 2 throughout. Note that for static node locations, we use $\mathbf{r}_{j,k} = \mathbf{r}_j = (r_j, s_j)$ by dropping the time index since static node locations do not change over time.

### D. Node mobility model

When directed at time $k$, mobile node $j$ moves with a speed of $v_{j,k} \in [v_{min}, v_{max}]$ in a direction $\theta_{j,k} \in [0, 2\pi)$. The node location $(r_{j,k+1}, s_{j,k+1})$ of the $j$-th mobile node at time $k+1$ is given by $r_{j,k+1} = r_{j,k} + T_s v_{j,k} \cos\theta_{j,k}$ and $s_{j,k+1} = s_{j,k} + T_s v_{j,k} \sin\theta_{j,k}$ for $j \in \mathcal{V}_m$.

## III. Distributed cluster Based Target Tracking by Particle Filters

In the following we propose a cluster-based MAT algorithm for mobile target tracking in the hybrid sensor network. We assume that there are few static nodes with high processing capabilities which act as cluster heads. The cluster head formation can be performed at the deployment stage, for example, based on Voronoi partitions. Although a cluster has a fixed number of static nodes, depending on the mobile nodes' mobility, the number of mobile nodes belonging to a particular cluster may change over time. In the proposed tracking algorithm, we assume that each cluster head keeps track on the mobile nodes entering and leaving the corresponding cluster at each time.

Let $C$ be the total number of clusters (or cluster heads) in the network and $n_{c,k}$ be the number of total nodes belongs to the cluster $c$ at time $k$, and $\mathcal{V}_c^k$ be the set containing $n_{c,k}$ number of these nodes, for $c = 1, \cdots, C$. The active cluster head at a given time is selected as the closest one to the predicted target position at that time. In the cluster based approach, each node belonging to the cluster $c$, sends its observation to the cluster head of the $c$-th cluster, $CH_c$ for $c = 1, 2, \cdots, C$. Since nodes have to communicate only with their cluster heads, a significant reduction of transmit power can be achieved compared to that with the centralized approach. On the other hand, the cluster head performing the tracking task is changing over time according to the predicted target locations. Thus one cluster head does not have to be active all the time. We assume that the observations are sent to the cluster head over AWGN channel. The received signal at the cluster head $CH_c$ from the $j$-th node at time $k$ is,

$$y_{j,k} = z_{j,k} + \epsilon_{j,k},\ \text{ for } j \in \mathcal{V}_c^k, \tag{3}$$

where $\epsilon_{j,k}$ is the received noise which is assumed to be Gaussian with mean zero and the variance $\sigma_\epsilon^2$. Denote $\mathbf{y}_{c,1:k} = [y_{1,1:k}, y_{2,1:k}, \cdots, y_{n_{c,k},1:k}]$ the observation vector at $CH_c$ up to time $k$. Since the observation vector at the cluster head $CH_c$ is non-linear function of the state to be estimated, and non-Gaussian if the local observations at nodes are quantized, we propose to use *sampling importance re-sampling* (SIR) particle filter [11] at the active cluster head. Compared to other variations of particle filters, SIR filter is more convenient to implement [11]. According to SIR filter, the particles are generated from the state transition probability: $\mathbf{x}_k^i \sim p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)$ where $\mathbf{x}_k^i$ is the $i$-th particle at time $k$, and the corresponding weights are updated according to $w_k^i \propto p(\mathbf{y}_{c,k}|\mathbf{x}_k^i)$ where $\mathbf{y}_{c,k}$ is the observation vector at time $k$ at $CH_c$.

From (2) and (3), the conditional pdf $p(\mathbf{y}_{c,k}|\mathbf{x}_k)$ for the $c$-th cluster head is given by, $p(\mathbf{y}_{c,k}|\mathbf{x}_k) = \mathcal{N}(\tilde{\mathbf{a}}_{c,k}, \tilde{R}_{c,k})$ where $\tilde{\mathbf{a}}_{c,k} = [a_{1,k}, a_{2,k}, \cdots, a_{n_{c,k},k}]^T$ is an $n_{c,k}$-length vector with $a_{j,k} = \frac{A_0}{|\mathbf{r}_{j,k} - \tilde{\mathbf{x}}_k|^{\alpha/2}}$, $for\ j = 1, 2, \cdots, n_{c,k}$, and $\tilde{R}_{c,k} = (\sigma_\nu^2 + \sigma_\epsilon^2)\mathbf{I}_{c,k}$ where $\mathbf{I}_{c,k}$ is the $n_{c,k} \times n_{c,k}$ identity matrix.

Then the weight update of the SIR filter at $CH_c$ at time $k$ can be performed as, $w_{c,k}^i \propto p(\mathbf{y}_{c,k}|\mathbf{x}_k^i) = \mathcal{N}(\tilde{\mathbf{a}}_{c,k}(\mathbf{x}_k^i), \tilde{R}_{c,k})$.

Denote $\hat{\mathbf{x}}_{c,k|k}$ and $U_{c,k|k}$ to be the state estimator and the error covariance matrix at the $c$-th cluster head at time $k$. Then the estimator and the error covariance matrix at the cluster head $CH_c$ are given by, $\hat{\mathbf{x}}_{c,k|k} \approx \sum_{i=1}^{S} w_{c,k}^i \mathbf{x}_k^i$ and $U_{c,k|k} \approx \sum_{i=1}^{S} w_{c,k}^i (\mathbf{x}_k^i - \hat{\mathbf{x}}_{c,k|k})(\mathbf{x}_k^i - \hat{\mathbf{x}}_{c,k|k})^T$, respectively. The predicted estimator and the error covariance matrices are given by $\hat{\mathbf{x}}_{c,k+1|k} = \mathbf{F}\hat{\mathbf{x}}_{c,k|k}$ and $U_{c,k+1|k} = \mathbf{F}U_{c,k|k}\mathbf{F}^T + Q$.

Once the estimator is computed at time $k$ at $CH_c$, the cluster head that would perform the tracking task at time $k+1$ is selected based on the predicted position estimate $\hat{\hat{\mathbf{x}}}_{c,k+1|k}$ where $\hat{\hat{\mathbf{x}}}_{c,k+1|k}$ represents the first two elements of $\hat{\mathbf{x}}_{c,k+1|k}$. Denote $\mathbf{r}_c$ to be the location of the $c$-th cluster head for $c = 1, \cdots, C$. Then, the cluster head that should perform the tracking task at time $k+1$, is $\hat{CH} = \underset{CH_i, i \in \mathcal{I}_c}{argmin} |\hat{\hat{\mathbf{x}}}_{c,k+1|k} - \mathbf{r}_i|$, where $\mathcal{I}_c$ is the set of indices of the neighboring cluster heads of the cluster head $CH_c$ including itself. Let $\hat{CH} = CH_d$. If the selected $CH$, $CH_d \neq CH_c$, then, $CH_c$ transmits its estimator to $CH_d$. Then for the particle filtering, $CH_d$ samples particles from $p(\mathbf{x}_{k+1}|\mathbf{x}_k = \hat{\mathbf{x}}_{c,k|k})$ and weight updating is performed based on the observation likelihood at $CH_d$ at time $k+1$.

## IV. Node mobility management

In the proposed MAT algorithm, the idea is to maintain a continuous $\beta$-coverage (exact or approximate) on the predicted trace of the moving target's trajectory at each time $k$. We call a point $\mathbf{r}_0$ is *$\beta$-covered* if there is at least $\beta$-number of sensors located within the disk denoted by $D(\mathbf{r}_0, r_D)$ centered at $\mathbf{r}_0$ with a radius of $r_D$ where $r_D$ is a design parameter.

Note that with the assumed observation model (2), the signal strength emitted by the target decays as the distance from the target is increasing. Thus the sensor nodes located closer to the target position at a given time make rich observations while those located far away from the current target location make poor observations. To better track the target at every time step, it is important to maintain a certain number of nodes very close to the target location at each time such that they receive rich observations. Let $r_D$ be the distance from the target where we call the nodes within this distance receive rich observations. $r_D$ can be selected such that the signal strength received at any node located within the disk $D(\mathbf{r}_0, r_D)$ from a target when the target is located at $\mathbf{r}_0$, exceeds a certain threshold. One straight-forward way of maintaining a $\beta$-coverage with a static network is to deploy a large number of nodes in the desired region such that each point in the region is covered by $\beta$-number of nodes. However, when it is necessary to cover a large region, to achieve this a large number of static nodes may be required. On the other hand, after the initial deployment if the nodes become inactive (due to node power failure or node breakage, etc..) the required $\beta$-coverage at each point cannot be achieved.

Denote the predicted target location at time $k$ by $\mathbf{P}_{k+1|k}$. When $\mathbf{P}_{k+1|k}$ is covered by at least $\beta$-number of static nodes, (i.e. there is a $\beta$ number of static nodes in the disk $D(\mathbf{P}_{k+1|k}, r_D)$ as shown in Fig. 1) we say *exact* $\beta$-coverage is achieved at $\mathbf{P}_{k+1|k}$. If $\mathbf{P}_{k+1|k}$ is not covered by $\beta$-number of static nodes, the required number of mobile nodes are directed to move the minimum distance needed to provide a $\beta$-coverage. At time $k$, if all the required number of mobile nodes can reach the desired destination to provide $\beta$-coverage (i.e. they can move such that after $T_s$ time they can be within

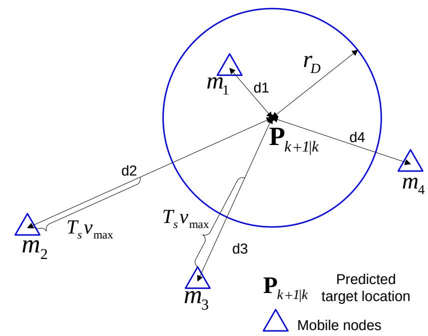


Fig. 1. Illustration of *exact* and *approximate* $\beta$ coverage

the disk $D(\mathbf{P}_{k+1|k}, r_D)$), we call the position $\mathbf{P}_{k+1|k}$ achieves *exact* $\beta$-coverage (as mobile node $m_3$ in Fig. 1). At sometimes, due to speed limitations of mobile nodes, some mobile nodes may not be able to reach the disk $D(\mathbf{P}_{k+1|k}, r_D)$ even if they move with their maximum speed to provide $\beta$-coverage (as mobile node $m_2$ in Fig. 1). However, in this case, since they move the maximum distance they can move towards the predicted target location so that the signal strength received at their destination is higher than that of the original location. If this happens, we call an *approximate* $\beta$-coverage is achieved at the predicted target location. We further assume that the current cluster head communicates with neighboring cluster heads to get location information of near-by mobile nodes to the predicted target position (outside its own cluster) if the current cluster head does not have sufficient number of mobile nodes to provide the required $\beta$-coverage.

The proposed node mobility management scheme has 4 basic steps:

- The active cluster head checks the number of static nodes $n_{s,D}$ within the disk $D(\hat{\hat{\mathbf{x}}}_{c,k+1|k}, r_D)$ where $\hat{\hat{\mathbf{x}}}_{c,k+1|k}$ is the predicted position of the target at cluster head $CH_c$, as before. If $n_{s,D} \geq \beta$, mobile nodes in the corresponding cluster remain stationary.
- If $n_{s,D} < \beta$, the difference $(\beta - n_{s,D})$ is determined.
- Determine $(\beta - n_{s,D})$ number of mobile nodes which should be directed to move. If the current cluster does not have $(\beta - n_{s,D})$ number of mobile nodes, the cluster head communicates with neighboring clusters (which are located close to the predicted target location) to determine the required number of mobile nodes needed from neighboring clusters.
- Determine the speed and direction of these selected mobile nodes.

Let $CH_d$ be the candidate cluster head for time $k+1$ which is selected at time $k$ and $CH_c$ be the active cluster head which performs the tracking task at time $k$. Once $CH_c$ determines the predicted location $\hat{\hat{\mathbf{x}}}_{c,k+1|k}$, it is transmitted to $CH_d$ (if it is different from $CH_c$). Note that if $\hat{\hat{\mathbf{x}}}_{c,k+1|k} \in CH_c$, then $CH_d = CH_c$. The cluster head $CH_d$ is responsible for managing mobile node mobility to maintain $\beta$-coverage for the predicted location, before start making measurements. Let $n_{m,d}^k$ be the total number of mobile nodes belonging to $CH_d$ at time $k$ and $\mathcal{V}_{m,d}^k$ be the set containing corresponding mobile node indices. If $n_{s,D} < \beta$, the cluster head $CH_d$

selects $(\beta - n_{s,D})$ number of mobile nodes which are closest to $\hat{\hat{\mathbf{x}}}_{c,k+1|k}$. If $(\beta - n_{s,D}) > n^k_{m,d}$, $CH_d$ selects closest $n^k_{m,d} - (\beta - n_{s,D})$ number of mobile nodes from neighboring clusters (located closer to the predicted target location) by communicating locally with neighboring cluster heads. Denote $\tilde{\mathcal{V}}^k_{m,d}$ be the set containing indices of $(\beta - n_{s,D})$ number of mobile nodes which are closest to $\hat{\hat{\mathbf{x}}}_{c,k+1|k}$. Note that $\tilde{\mathcal{V}}^k_{m,d} \subseteq \mathcal{V}^k_{m,d}$ only if $n^k_{m,d} \geq (\beta - n_{s,D})$. According to the assumption, a mobile node $j$ can move in a direction $\theta_{j,k} \in [0, 2\pi)$ with a speed of $v_{j,k} \in [v_{min}, v_{max}]$ from time $k$ to $k+1$. Now the objective is to determine the best $\theta_{j,k}$ and $v_{j,k}$ for $j \in \tilde{\mathcal{V}}^k_{m,d}$ such that corresponding mobile nodes move the minimum distance to provide a $\beta$-coverage for $\hat{\hat{\mathbf{x}}}_{c,k+1|k}$ at time $k+1$. The best $\theta_{j,k}$ and $v_{j,k}$ for the $j$-th mobile node for $j \in \tilde{\mathcal{V}}^k_{m,d}$ are given as follows: If $(|\mathbf{r}_{j,k} - \hat{\hat{\mathbf{x}}}_{c,k+1|k}| - r_D) > 0$ and $(|\mathbf{r}_{j,k} - \hat{\hat{\mathbf{x}}}_{c,k+1|k}| - r_\beta) \leq v_{max}T_s$,

$$\hat{v}_{j,k} = \max\left\{\frac{1}{T_s}|\mathbf{r}_{j,k} - \hat{\hat{\mathbf{x}}}_{c,k+1|k}|, v_{min}\right\} \quad (4)$$

$$\hat{\theta}_{j,k} = atan2\left(\frac{s_{j,k} - \hat{x}_{c,2(k+1|k)}}{r_{j,k} - \hat{x}_{c,1(k+1|k)}}\right) \quad (5)$$

If $(|\mathbf{r}_{j,k} - \hat{\hat{\mathbf{x}}}_{c,k+1|k}| - r_D) > 0$ and $(|\mathbf{r}_{j,k} - \hat{\hat{\mathbf{x}}}_{c,k+1|k}| - r_D) > v_{max}T_s$,

$$\hat{v}_{j,k} = v_{max} \quad (6)$$

$$\hat{\theta}_{j,k} = atan2\left(\frac{s_{j,k} - \hat{x}_{c,2(k+1|k)}}{r_{j,k} - \hat{x}_{c,1(k+1|k)}}\right) \quad (7)$$

If $(|\mathbf{r}_{j,k} - \hat{\hat{\mathbf{x}}}_{c,k+1|k}| - r_D) < 0$, then $\hat{v}_{j,k} = 0$ where $atan2(x)$ is the four quadrant inverse tangent of $x$, and $\hat{x}_{c,1(k+1|k)}$ and $\hat{x}_{c,2(k+1|k)}$ are the $X$ and $Y$ coordinates of the predicted target location $\hat{\hat{\mathbf{x}}}_{c,k+1|k}$. Details are omitted here due to space limitations.

The pseudocode for the proposed node mobility algorithm at time $k$ is given in Algorithm 1.

**Algorithm 1** Node mobility management algorithm for MAT

INPUT: Predicted target location: $\hat{\hat{\mathbf{x}}}_{c,k+1|k}$, number of static nodes for $CH_d$: $n^k_{s,d}$, number of mobile nodes for $CH_d$ at time $k$: $n^k_{m,d}$

OUTPUT: Optimal speed $\hat{v}_{j,k}$ and the direction $\hat{\theta}_{j,k}$ of $j$-th mobile node for $j \in \tilde{\mathcal{V}}^k_{m,d}$

PROCEDURE:

```
Find number of static nodes inside the disk (x̂̂_{c,k+1|k}, r_D), n_{s,D},
Check → n_{s,D} ≥ β [i.e. x̂̂_{c,k+1|k} is β-covered by static nodes]
if yes (i.e. n_{s,D} ≥ β) then
    r_{j,k+1} = r_{j,k} for j ∈ V^k_{m,d} ⇒ no mobile node needs to move
else {no (i.e. n_{s,D} < β )}
    Check → n^k_{m,d} ≥ (β − n_{s,D}) (i.e. to check whether CH_d has sufficient mobile nodes)
    if yes then
        Find Ṽ^k_{m,d} from V^k_{m,d}
    else {no}
        Communicate with local CHs to find mobile node locations in neighboring clusters to form the set Ṽ^k_{m,d} as described in Section IV.
    end if
end if
for j = 1 : size(Ṽ^k_{m,d}) do
    f(j) = |r_{j,k} − x̂̂_{c,k+1|k}|
    if f(j) − r_D ≤ 0 then
        r_{j,k+1} = r_{j,k}
    else {f(j) − r_D > 0 & f(j) − r_D > T_s v_max}
        Compute v̂_{j,k} and θ̂_{j,k} from (6) and (7)
    else {f(j) − r_D > 0 & f(j) − r_D < T_s v_max}
        compute v̂_{j,k} and θ̂_{j,k} from (4) and (5)
    end if
end for
```

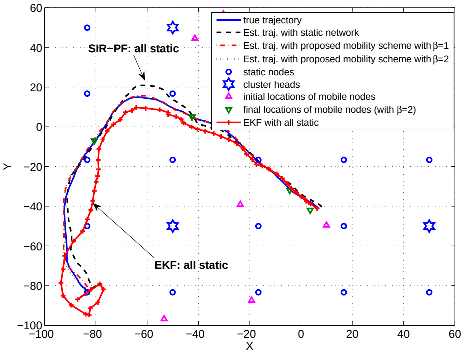


Fig. 2. Estimated trace of the target trajectory with proposed MAT algorithm; $n_s = 36$, $n_m = 12$, $C = 4$, $v_{max} = 10ms^{-1}$, $r_D = 5m$

## V. Performance Analysis

For the performance analysis we assume a sensor network deployed in a square region with area of $200m \times 200m$. The network is assumed to be consisting of 4 cluster heads and the clustering is performed based on Voronoi partitions. We assume that there are 36 number of static nodes (including cluster heads) are deployed in a grid. The initial target state is assumed to be Gaussian with mean $\boldsymbol{\mu}_0$ and covariance matrix $\boldsymbol{\Sigma}_0$. We assume $\boldsymbol{\mu}_0 = [-80 \quad -80 \ 1 \ 1]^T$ and $\boldsymbol{\Sigma}_0 = diag([10 \ 10 \ 0.1 \ 0.1]^T)$. Sampling time is assumed to be $T_s = 1s$. The intensity of the state process noise $\sigma^2_u = 0.4$. Observation noise variances at individual nodes, and cluster heads, $\sigma^2_\nu$ and $\sigma^2_\epsilon$ are set to 0.1. The target amplitude $A_0 = 100$. The tracking is performed for $60s$ and the number of particles in the particle filter is set to $S = 1000$. The performance measure is taken as the root mean square error (RMSE) of the target position estimate given by, $RMSE_k = \sqrt{((x_k - \hat{x}_k)^2 + (y_k - \hat{y}_k)^2)}$. The RMSE is compared with the square root of the Posterior Cramér-Rao lower bounds (PCRLB) components of the position error. We omit the analysis of PCRLB here due to space limitations. Figures 2 and 3 show the performance of the proposed MAT scheme when there are 12 mobile nodes in the network. In both figures we assume that $r_D = 5m$ and $v_{max} = 10ms^{-1}$. In Fig. 2 estimated and true trajectories are shown with the assumed parameters and for different $\beta$. It can be seen that by allowing 3 nodes per cluster to be mobile, the target trajectory can be tracked with high accuracy compared to that with the scenario where all nodes are static. For the static network performance, we assume that all mobile nodes make measurements at their initial locations. The results in Figs 2 and 3 are averaged over 50 Monte Carlo trials. In Fig. 2, we also compare the SIR-PF based results with the results obtained via extended Kalman filter (EKF) in the case of when all nodes are static. We observed 20 lost tracks out of 50 in the case of EKF and 0 for SIR-PF with all static nodes. Thus in our problem, SIR-PF performs well compared to EKF.

The RMSE and PCRLB analysis for proposed MAT scheme for $v_{max} = 10ms^{-1}$ and $r_D = 5m$ is shown in Fig. 3. The results are shown for $\beta = 1$ and $\beta = 2$. It can be seen that when the objective is to maintain at least $\beta = 2$ number of nodes within the disk $D(\hat{\hat{\mathbf{x}}}_{c,k+1|k}, r_D = 5m)$, the target trajectory can be tracked with lower RMSE. Also in

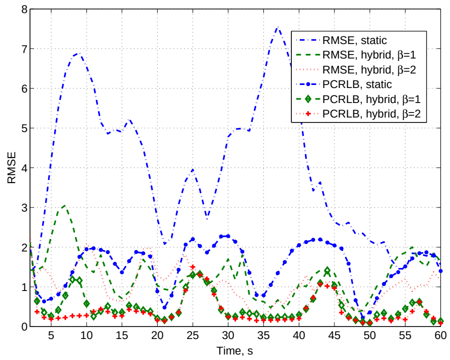


Fig. 3. RMSE and PCRLB (in $m$) for the estimated target position when $\beta$ is changing; $n_s = 36$, $n_m = 12$, $C = 4$, $v_{max} = 10ms^{-1}$, $v_{min} = 0$, $r_D = 5m$

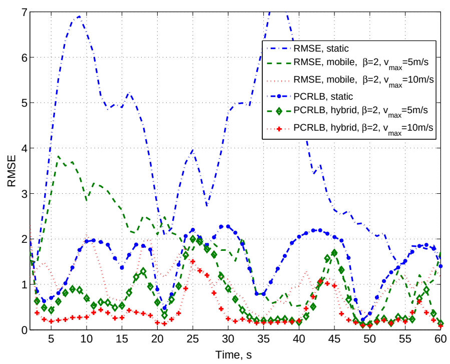


Fig. 4. RMSE and PCRLB (in $m$) for the estimated target position when $v_{max}$ is varying; $n_s = 36$, $n_m = 12$, $C = 4$, $v_{min} = 0$, $r_D = 5m$, $\beta = 2$

that case, it can be seen that the RMSE performance gets very closed to the derived Posterior Cramer-Rao lower bound. For $\beta = 1$, that is to maintain at least 1 node within the disk $D(\hat{\hat{\mathbf{x}}}_{c,k+1|k}, r_D = 5m)$ at each $k$, a considerable performance gain can be achieved compared to that with the static network. Note that there is always a trade-off between the value $\beta$ and the energy consumption of mobile nodes, since when $\beta$ is getting larger the number of mobile nodes to be moved is also increasing although it provides a high performance gain. On the other hand, it is of interest to investigate the performance metrics, when the maximum speed of a mobile node is varying. In the next experiment, we investigate the effect of the maximum node speed for the proposed MAT algorithm when $r_D$ is fixed. Figure 4 depicts the performance metrics when the maximum speed of a mobile node is varying. The results in Fig. 4 are corresponding to $v_{max} = 5m/s$ and $v_{max} = 10m/s$, and $\beta = 2$. It can be seen that for low values of $v_{max}$ the performance gain is quite decreasing compared to higher values of $v_{max}$. This is due to the fact that for lower $v_{max}$ values the maximum distance that a mobile node can move from one time period is lower, thus almost it might provide an approximate $\beta$-coverage rather than exact $\beta$-coverage. Also, with lower $v_{max}$, a considerable performance gain is achieved compared to all-static network, for a given $r_D$. However, as mentioned earlier, there is always a trade-off among the parameters $v_{max}$, $\beta$, $r_D$ and the required performance gain.

Although figures are not included due to space limitations, it can be seen that when the number of mobile nodes is large, the tracking performance with a given $\beta$ ($\beta = 2$) does not have much effect on the maximum speed of mobile nodes. This is due to the fact that when there is a relatively large number of nodes, it is more likely that there is a sufficient number of nodes around the required position and these nodes can reach the required location by moving small distances. Also it is observed that for a given number of static nodes, when the number of mobile nodes is increasing, the average total distance that a mobile nodes has to move to provide the required $\beta$-coverage dynamically, is significantly reduced. This is because, for large $n_m$, it has more flexibility to find mobile nodes in the close proximity of the predicted target position which would provide the desired $\beta$ coverage by moving a small distance at any given time.

## VI. Conclusions

This paper proposed a novel cluster based mobility assisted target tracking algorithm exploiting node mobility in a hybrid sensor network consisting of both static and mobile nodes. In the proposed MAT algorithm, node mobility is exploited to maintain a desired *coverage level* on the trace of the target at each time dynamically as the target moves. It can be seen that when the mobile nodes are directed to move to achieve relatively a higher *coverage level*, RMSE performance of the target position estimate is getting much more closer to PCRLB even with a relatively small number of mobile nodes. Also, since mobile nodes are directed to move only to compensate for the lack of coverage resulted by static nodes, continuous mobility is not required saving mobile node locomotion energy. The proposed scheme is robust against cluster head or node failures in the network.

## References


[1] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, “Particle filters for positioning, navigation, and tracking,” *IEEE Trans. on Signal Processing,*, vol. 50, no. 2, pp. 425 – 437, Feb. 2002.

[2] P. M. Djurić, M. Vemula, and M. F. Bugallo, “Target tracking by particle filtering in binary sensor networks,” *IEEE. Trans. on Signal Processing*, vol. 56, no. 6, pp. 2229–2238, June 2008.

[3] O. Ozdemir, R. Niu, and P. K. Varshney, “Tracking in wireless sensor networks using particle filtering: Physical layer considerations,” *IEEE. Trans. on Signal Processing*, vol. 57, no. 5, pp. 1987–1999, May 2009.

[4] M. Coates, “Distributed particle filters for sensor networks,” in *Third Int. Symp. on Information Processing in Sensor Networks, (IPSN)*, Berkeley, CA, Apr. 2004, pp. 99–107.

[5] A. S. Bashi, V. P. Jilkov, X. R. Li, and H. Chen, “Distributed implementations of particle filters,” in *Proc. of the Sixth Int. Conf. of Information Fusion*, Cairns, Autralia, July 2003, pp. 1164–1171.

[6] Y. Zou and K. Chakrabarty, “Distributed mobility management for target tracking in mobile sensor networks,” *IEEE. Trans. on Mobile Computing*, vol. 6, no. 8, pp. 872–887, Aug. 2007.

[7] R. Olfati-Saber, “Distributed tracking for mobile sensor networks with information-driven mobility,” in *Proc. of American Control Conf.*, New York City, NY, July 2007, pp. 4606–4612.

[8] Y. Li and P. M. Djurić, “Particle filtering for target tracking with mobile sensors,” in *Proc. Acoust., Speech, Signal Processing (ICASSP)*, Honolulu, Hawai’i, U.S.A., Apr. 2007, pp. II–1101–II–1104.

[9] ——, “Target tracking with mobile sensors using cost-reference particle filtering,” in *Proc. Acoust., Speech, Signal Processing (ICASSP)*, Las Vegas, Nevada, U.S.A., Apr. 2008, pp. 2549–2552.

[10] A. S. Chhetri, D. Morrell, and A. Papandreou-Suppappola, “Scheduling multiple sensors using particle filters in target tracking,” in *IEEE Workshop on Statistical Signal Processing*, Sept. 2003, pp. 549– 552.

[11] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” *IEEE. Trans. on Signal Processing*, vol. 50, pp. 174–188, Feb. 2002.