The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

# Tracking Point or Diffusing Targets Using Mobile Sensor Networks under Sensing Noises

Ying-ying Li and Yun-hui Liu
*Department of Mechanical and Automation Engineering*
*The Chinese University of Hong Kong, NT, Hong Kong*
{yyli,yhliu}@mae.cuhk.edu.hk

*Abstract*—**This paper presents a distributed algorithm for a mobile sensor network to track targets with unknown motion. We formulates the target tracking as a multi-objective optimization problem which integrates the tracking quality, the energy saving and the network connectivity. To cope with sensing noises, we use the determinant of the covariance matrix of target estimation as the tracking quality measure and compute its partial derivatives for the optimization process. Virtual nodes are introduced to represent obstacles in the environment. Furthermore this algorithm can be extended to solve the problem of source tracking where sensors can only detect the density of the diffusing substances emitted by the source. Therefore a whole tracking framework has been set up which can be easily extended for applications under complicated situations. Simulations demonstrate the effectiveness of the proposed algorithm in energy conservation and tracking accuracy under different situations.**

## I. Introduction

Due to attractive characteristics such as wide coverage of environment, fast response to changes and high reliability for information gathering, mobile sensor networks have been widely used in civil and military applications, such as surveillance, environmental control, and health care. Target tracking using a mobile sensor network is to maintain the moving targets within the sensing coverage of the network by properly moving the nodes.

This topic is being extensively studied in robotics and sensor networks in recent years and several approaches have been developed. Makarenko [1], Chung [2] and Spletzer [3] proposed to optimize positions of the sensor nodes for tracking under different performance index. Jung [4] developed an approach distributing the robots in the region according to the target densities. The aforementioned methods determine node motion on the basis of the data collected by all the sensor nodes under the assumption that the communication link always exists between any two nodes during tracking. This communication requirement can hardly be satisfied in mobile sensor networks with a large number of nodes. Parker [5] combined virtual forces with high-level behavior-based probabilities for controlling the motion of robots. Shucker [6] simplified the network into a virtual spring mesh and used the incident edges of nodes to generate the control force. In those works, it was assumed that accurate information about the target is available, which is not true in real applications. Zou [7] integrated the mobility-enhanced improvement with negative consequences and proposed a distributed mobility management scheme.

Almost all these works assumed that there is no sensing noise or obstacle in the environment. Moreover, they all coped with point targets whose position can be directly detected by sensors. There is no work on emission source tracking where we need to find the source through tracking its diffusing substances. And for sensor networks with scarce energy resources, it is not favorable for nodes to move because the energy required for locomotion energy is much higher than that for sensing and communication.

This paper extends our previous work [8] to tracking point targets and emission sources with the existence of sensing noises and obstacles. The algorithm developed was for tracking a point target with unknown motion while minimizing the energy consumption and maintaining network connectivity, provided that the sensor measurements are accurate. In this paper, we cope with sensing noises in the measurements and consider both point targets and emission source tracking. For sensing noises, the determinant of the covariance matrix of the target estimation is adopted as the tracking quality measure. Then based on the calculus rule of matrix, we compute its partial derivatives to be used in on-line optimization process. For obstacle avoidance, we incorporate virtual nodes in the potential function to represent the obstacle existence. For emission source tracking, we use the product integral of the point tracking quality and the substance density over the whole area as the new tracking quality measure. Then the previous tracking strategy can be adopted here. We can see that a whole tracking framework has been set up which can be easily modified and extended for many complicated situations in reality. Simulations have been conducted and the results confirmed good performance of the algorithm.

## II. Algorithm Outline

We first describe some general notations and models that will be applied throughout in this entire paper. Then the proposed algorithm in our previous paper will be outlined.

### A. Notations and Modeling

Consider a mobile sensor network with a large number of sensor nodes deployed on a 2-D plane. There may be point targets with unknown motion or sources that emit diffusing substances around. For point target tracking, the target can be detected by the nodes. Then the problem is to trace the target by moving the nodes so that

This work is affiliated with the Microsoft-CUHK Joint Laboratory for Human-centric Computing and Interface Technologies. This work is in part supported the Hong Kong RGC under grant 414505.

the target is always within the sensing coverage of the network. For emission source tracking, the source may be locomotive and hidden. Its location cannot be detected directly by the nodes. But its emitted substances diffuse around continually and the nodes can measure the properties of these substances such as temperature, density, etc. Then we can approximate the source location through tracking the substances because the source always exists at the point with the maximal density of the substance.

The notations used in this paper are given in table I. We use bold letters to represent vectors. To clarify the tracking problem, some assumptions are made as follows:

*i)* The communication region of each node is modeled as circular discs with the same communication radius $r_c$ .

*ii)* The sensor network is abstracted into a graph $G=(V,E)$ . $V$ is the set of nodes. The edge $(i,j)\in E$ connects node $i$ to node $j$ if $i$ is in the communication region of $j$ .

*iii)* The initial network is connected but the whole network topology is not available.

*iv)* Every node knows its position accurately.

TABLE I. NOTATIONS USED THROUGHOUT THE PAPER

| Notation | Description |
|---|---|
| $k$ | The $k$-th sampling time |
| $\Delta t$ | The sampling time interval |
| $\boldsymbol{\eta}_i(k)$ | The location of node $i$ at time $k$ |
| $\mathbf{z}_i(k)$ | The measurement of node $i$ at time $k$ |
| $\mathbf{u}_i(k)$ | the control input of node $i$ at time $k$ |
| $\boldsymbol{\eta}_T(k)$ and $\dot{\boldsymbol{\eta}}_T(k)$ | The location and velocity of the target at time $k$ |
| $\hat{\boldsymbol{\eta}}_T(k)$ | The target location estimation at time $k$ |
| $\boldsymbol{\mu}_i(k)$ and $\mathbf{P}_i(k)$ | The mean value and the covariance matrix of the target location estimation on node $i$ at time $k$ |
| $\rho(\mathbf{q},k)$ | The substance density at point $\mathbf{q}$ at time $k$ |
| $j_1,j_2,\ldots\in N(i,k)$ | The one-hop RNG neighbors of node $i$ at time $k$ |

At every $k$, each node updates its RNG neighbor list $N(i,k)$ and maintains connectivity only with them during the motion within current time interval. Each node is subject to the same motion model as

$$\boldsymbol{\eta}_i(k)=\boldsymbol{\eta}_i(k-1)+\mathbf{u}_i(k) \tag{1}$$

Assume the target is initially within the sensing coverage of the network. For small sampling time interval, it is possible to adopt linear model to model the target motion

$$\boldsymbol{\eta}_T(k)=\boldsymbol{\eta}_T(k-1)+\Delta t\cdot\dot{\boldsymbol{\eta}}_T(k-1)+\mathbf{w}(k-1) \tag{2}$$

$\mathbf{w}(k)$ represents the white noise with zero mean. The introduction of the noise term is to compensate for the high order motion. The target can be observed by the sensors equipped on the nodes. The sensing model of the target by sensors on node $i$ is given by:

$$\mathbf{z}_i(k)=h_i(\boldsymbol{\eta}_T(k),\boldsymbol{\eta}_i(k))+\mathbf{v}(k) \tag{3}$$

$h_i()$ is the sensing function of node $i$, which means the sensing model of nodes can be different. $\mathbf{v}(k)$ represents the white noises with zero mean. Assume that the distributions of noises $\mathbf{v}(k)$ and $\mathbf{w}(k)$ are $N(0,\mathbf{W})$ and $N(0,\mathbf{V})$ respectively.

*Distributed target tracking problem*: Given a mobile sensor network and a moving target with unknown motion, design an algorithm such that the target is maintained within the sensing coverage of the network and:

*i)* The network maintains connected during the tracking.

*ii)* The number of nodes that must be moved and their motion steps must be minimized to minimize the energy consumption.

*iii)* The algorithm must be distributed, i.e. each node must determine its motion only by either its own information or the information of its one-hop neighbors.

### B. Distributed Tracking Algorithm Outline

In [8], we have proposed a distributed tracking algorithm in which the nodes are classified into idle and active nodes. During the tracking process, the active nodes will move and the idle nodes will remain at their current positions. The nodes switch between active and idle state so that the tracking becomes a node-to-node hand-off process as the target moves through the region.

At every time, each active node estimates and predicts the target motion. It optimizes its motion using a heuristic algorithm under three criteria. First, the target escaping probability is defined to measure the tracking quality. Second, the potential function is defined to represent the network connectivity status as

$$U(\boldsymbol{\eta}(k))=\sum_{i=1}^{n}\sum_{j\in N(i,k)}\phi(\|\boldsymbol{\eta}_i(k)-\boldsymbol{\eta}_j(k)\|) \tag{4}$$

where $\phi()$ is the function which represents the distance constraint between two neighbors. Third, the kinetic energy of the nodes is determined by the node velocity during $\Delta t$

$$C(\boldsymbol{\eta}(k))=\sum_{i=1}^{n}\frac{1}{2\Delta t^2}m_i\|\boldsymbol{\eta}_i(k)-\boldsymbol{\eta}_i(k-1)\|^2 \tag{5}$$

We propose a local motion strategy to solve the multi- objective optimization problem

$$\max_D\{Q(\boldsymbol{\eta}(k)),-C(\boldsymbol{\eta}(k)),U(\boldsymbol{\eta}(k))\} \tag{6}$$

Based on the multi-objective optimization theory, it is transformed to a single objective optimization as

$$\max_D J(\boldsymbol{\eta}(k))=\max_D[Q(\boldsymbol{\eta}(k))+\alpha U(\boldsymbol{\eta}(k))-\beta C(\boldsymbol{\eta}(k))] \tag{7}$$

where $\alpha$ and $\beta$ are the coefficients to unify the magnitude order. $\beta$ reflects the importance of energy conservation. Then we adopt the optimization recursive theory [9] to solve it.

## III. TRACKING POINT TARGETS WITH SENSING NOISES

Section III and IV will improve our previous algorithm to be suitable for complicated situations such as noisy sensing model, obstacle avoidance and emission source tracking.

*A. Noisy Sensing Model*

Using a proper quality measure to plan motion of active nodes is crucial for the tracking performance of the sensor network. When defining the quality measure, two factors must be taken in account. First, the measure must represent the tracking quality. Second, each node must be able to calculate the quality measure using information collected by itself and its one-hop neighbors so as to conduct distributed computation. We propose to use the estimation accuracy of the target motion as the tracking quality measure. Certainly, the better can the target motion be estimated, the better is it covered by the sensor network. Under the quality measure, the active nodes will move so as to yield more accurate estimation of the target motion.

We linearize the measurement model (3) as follows:

$$\mathbf{z}_i(k) = \mathbf{H}_i(k)\boldsymbol{\eta}_T(k) + \mathbf{v}(k) \tag{8}$$

$\mathbf{H}_i(k)$ is the Jacobian matrix of $h_i(\boldsymbol{\eta}_T(k), \boldsymbol{\eta}_i(k))$. For example, the acoustic amplitude sensor is usually adopted in sensor network applications. Assuming that the sound source is a point source and sound propagation is lossless and isotropic, the amplitude measurement $\mathbf{z}_i(k)$ is related to $\boldsymbol{\eta}_T(k)$ as

$$\mathbf{z}_i(k) = \frac{a}{\left\|\boldsymbol{\eta}_T(k) - \boldsymbol{\eta}_i(k)\right\|} + \mathbf{v}(k) \tag{9}$$

where $a$ represents the physical characteristics of the sensor. Its Jacobian matrix has the following form

$$\mathbf{H}_i(k) = \nabla_{\boldsymbol{\eta}_T(k)} h_i(\boldsymbol{\eta}_T(k), \boldsymbol{\eta}_i(k)) = \frac{-a(\boldsymbol{\eta}_T(k) - \boldsymbol{\eta}_i(k))}{\left\|\boldsymbol{\eta}_T(k) - \boldsymbol{\eta}_i(k)\right\|^2} \tag{10}$$

At every time, each node uses Kalman Filtering (KF) to calculate the estimation $\boldsymbol{\mu}_i(k/k)$ of the target state and the covariance matrix $\mathbf{P}_i(k/k)$ which represent the distribution property of the estimation probability. The distributed KF equations are as follows

$$\begin{aligned} \boldsymbol{\mu}_i(k/k-1) &= \boldsymbol{\mu}_i(k-1/k-1) + \Delta t \cdot \dot{\boldsymbol{\mu}}_i(k-1/k-1) \\ \mathbf{P}_i(k/k-1) &= \mathbf{P}_i(k-1/k-1) + \mathbf{W} \end{aligned} \tag{11}$$

$$\begin{aligned} \mathbf{K}(k) &= \mathbf{P}_i(k/k-1)\mathbf{H}^T(k)\left(\mathbf{H}(k)\mathbf{P}_i(k/k-1)\mathbf{H}^T(k) + \mathbf{V}\right)^{-1} \\ \boldsymbol{\mu}_i(k/k) &= \boldsymbol{\mu}_i(k/k-1) + \mathbf{K}(k)\left(\mathbf{z}(k) - \mathbf{H}(k)\boldsymbol{\mu}_i(k/k-1)\right) \\ \mathbf{P}_i(k/k) &= \left(\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k)\right)\mathbf{P}_i(k/k-1) \end{aligned} \tag{12}$$

$\mathbf{z}(k) = \begin{pmatrix}\mathbf{z}_i(k) & \mathbf{z}_{j_1}(k) & \mathbf{z}_{j_2}(k) & \cdots\end{pmatrix}^T$, $\mathbf{H}(k) = \begin{pmatrix}\mathbf{H}_i(k) & \mathbf{H}_{j_1}(k) & \mathbf{H}_{j_2}(k) & \cdots\end{pmatrix}^T$.

$\mathbf{P}_i(k/k)$ is a measure for the estimation uncertainty. Minimizing $\mathbf{P}_i(k/k)$ corresponds to minimizing the uncertainty. While entropy [10] contains more information than the covariance, it is at the expense of more computational costs and not suitable for low-configured nodes. Therefore, we choose the determinant of $\mathbf{P}_i(k/k)$ as the tracking quality measure of node $i$

$$Q(\boldsymbol{\eta}(k)) = -\det\left(\mathbf{P}_i(k/k)\right) \tag{13}$$

Since each node estimates the target location independently using KF based on the measurements of itself and its neighbors, so $Q(\boldsymbol{\eta}(k))$ can be certainly evaluated distributedly. And the measurements are determined by the locations of the nodes, so the independent variables of $Q(\boldsymbol{\eta}(k))$ are actually $\left(\boldsymbol{\eta}_i(k), \bigcup_{j \in N(i,k)} \boldsymbol{\eta}_j(k)\right)$.

To move the active nodes in the direction of minimizing the determinant of $Q(\boldsymbol{\eta}(k))$, it is necessary to calculate its partial derivative with respect to $\boldsymbol{\eta}_i(k)$. First we introduce the standard derivative calculus rule of matrix [11]. $\mathrm{f}: R^{n\times n} \to R$ is a real-valued matrix function. $\mathbf{A}(x) \in R^{n\times n}$ is a symmetric definite matrix with the scalar variable $x \in R$. $\mathbf{B}(x)$ is a matrix function with any dimension. $\mathbf{C} \in R^{n\times n}$ is a symmetric constant matrix. Then

$$\begin{aligned} \frac{d\,\mathrm{f}(\mathbf{A}(x))}{dx} &= \mathrm{tr}\left(\frac{d\,\mathrm{f}}{d\mathbf{A}}\frac{d\mathbf{A}(x)}{dx}\right) \\ \frac{d\left(\mathbf{A}(x)^{-1}\right)}{dx} &= -\mathbf{A}(x)^{-1}\left(\frac{d\mathbf{A}(x)}{dx}\right)\mathbf{A}(x)^{-1} \\ \frac{d\det(\mathbf{A})}{d\mathbf{A}} &= |\mathbf{A}|\,\mathbf{A}^{-1} \\ \frac{d\left(\mathbf{B}(x)^T\mathbf{C}\mathbf{B}(x)\right)}{dx} &= 2\mathbf{B}(x)^T\mathbf{C}\frac{d\mathbf{B}(x)}{dx} \end{aligned} \tag{14}$$

The partial derivative of $Q(\boldsymbol{\eta}(k))$ with respect to the $x$-coordinate of the location of node $i$ is given by

$$\begin{aligned} \frac{\partial Q(\boldsymbol{\eta}(k))}{\partial \boldsymbol{\eta}_{ix}(k)} &= -\mathrm{tr}\left(\frac{d\left(\det\left(\mathbf{P}_i(k/k)\right)\right)}{d\mathrm{P}_i(k/k)} \cdot \frac{\partial \mathbf{P}_i(k/k)}{\partial \boldsymbol{\eta}_{ix}(k)}\right) \\ &= -\mathrm{tr}\left(\left|\mathbf{P}_i(k/k)\right| \cdot \mathbf{P}_i(k/k)^{-1} \cdot \frac{\partial \mathbf{P}_i(k/k)}{\partial \boldsymbol{\eta}_{ix}(k)}\right) \\ &= -\left|\mathbf{P}_i(k/k)\right| \cdot \mathrm{tr}\left(\mathbf{P}_i(k/k)^{-1} \cdot \frac{\partial \mathbf{P}_i(k/k)}{\partial \boldsymbol{\eta}_{ix}(k)}\right) \end{aligned} \tag{15}$$

By substituting the term $\partial \mathbf{P}_i(k/k)/\partial \boldsymbol{\eta}_{ix}(k)$ by KF, we obtain

$$\begin{aligned} \frac{\partial \mathbf{P}_i(k/k)}{\partial \boldsymbol{\eta}_{ix}(k)} &= \frac{\partial\left(\left(\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k)\right)\mathbf{P}_i(k/k-1)\right)}{\partial \boldsymbol{\eta}_{ix}(k)} \\ &= \frac{\partial\begin{pmatrix}\mathbf{P}_i(k/k-1) - \mathbf{P}_i(k/k-1)\mathbf{H}(k)^T \\ \left(\mathbf{H}(k)\mathbf{P}_i(k/k-1)\mathbf{H}(k)^T + \mathbf{V}\right)^{-1}\mathbf{H}(k)\mathbf{P}_i(k/k-1)\end{pmatrix}}{\partial \boldsymbol{\eta}_{ix}(k)} \end{aligned} \tag{16}$$

$\mathbf{P}_i(k/k-1)$ is independent of $\boldsymbol{\eta}_i(k)$, so Eq. (16) is equal to

$$-\mathbf{P}_i(k/k-1) \cdot \frac{\partial\left(\mathbf{H}(k)^T\left(\mathbf{H}(k)\mathbf{P}_i(k/k-1)\mathbf{H}(k)^T + \mathbf{V}\right)^{-1}\mathbf{H}(k)\right)}{\partial \boldsymbol{\eta}_{ix}(k)} \cdot \mathbf{P}_i(k/k-1)$$

Using the matrix operations (14) and omitting $k$ and $k$-$1$ for simplification, the upper equation can be revised as

$$\begin{aligned} &= -\mathbf{P}_i\frac{\partial\left(\mathbf{H}^T\left(\mathbf{H}\mathbf{P}_i\mathbf{H}^T + \mathbf{V}\right)^{-1}\mathbf{H}\right)}{\partial \boldsymbol{\eta}_{ix}}\mathbf{P}_i \\ &= -\mathbf{P}_i\begin{bmatrix} 2\mathbf{H}^T\left(\mathbf{H}\mathbf{P}_i\mathbf{H}^T + \mathbf{V}\right)^{-1}\dfrac{\partial \mathbf{H}}{\partial \boldsymbol{\eta}_{ix}} \\ -2\mathbf{H}^T(\mathbf{H}\mathbf{P}_i\mathbf{H}^T + \mathbf{V})^{-1}\left(\mathbf{H}\mathbf{P}_i\dfrac{\partial \mathbf{H}^T}{\partial \boldsymbol{\eta}_{ix}}\right) \cdot \\ (\mathbf{H}\mathbf{P}_i\mathbf{H}^T + \mathbf{V})^{-1}\mathbf{H} \end{bmatrix}\mathbf{P}_i \end{aligned} \tag{17}$$

Similarly, it is possible to calculate the partial derivative of $Q(\boldsymbol{\eta}(k))$ with respect to the $y$-coordinate $\boldsymbol{\eta}_{iy}(k)$.

It is noted that $\boldsymbol{\eta}_T(k)$ is unavailable in the calculation of $\mathbf{H}_i(k)$ and $\mathbf{H}_j(k)$, so we use $\boldsymbol{\mu}_i(k/k-1)$ to approximate it. Since $\boldsymbol{\eta}_i(k)$ only affects the Jacobian matrix $\mathbf{H}_i(k)$, we have

$$\frac{\partial \mathbf{H}(k)}{\partial \boldsymbol{\eta}_i(k)} = \begin{pmatrix}\partial \mathbf{H}_i(k)/\partial \boldsymbol{\eta}_i(k) & 0 & \cdot & \cdot & \cdot & 0\end{pmatrix}^T \tag{18}$$

$|\mathbf{P}_i(k/k)|$ exists in both $x$ and $y$ derivatives, so it can be ignored in the determination of the node motion.

### B. Obstacle Avoidance

Usually many static or dynamic obstacles may exist in the field. We will add the capability of obstacle avoidance in the tracking framework and present an on-line avoidance planning for nodes while tracking the target. This algorithm is suitable for any type obstacle. As we know, $U(\boldsymbol{\eta}(k))$ represents the distance constraint between any two neighbors. If the obstacles are transformed to virtual neighbors of the nodes, the optimization process of $U(\boldsymbol{\eta}(k))$ will also keep nodes within the safe region of obstacles. At time $k$, node $i$ detects the obstacles $O_m(k)$. We suppose the obstacle surface is piecewise continuous and draw a vertical projection line from node $i$ to this surface. Then the projection points denote the virtual nodes as

$$\boldsymbol{\eta}'_{m,l}(k) = \arg\min_{\mathbf{q}\in O_m(k)} \left\| \mathbf{q} - \boldsymbol{\eta}_i(k) \right\|, l = 1,2,\cdots. \quad (19)$$

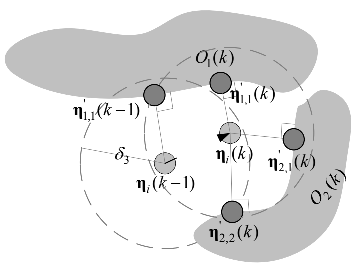


Fig.1. The virtual nodes corresponding to obstacles

Fig. 1 shows how to generate the virtual nodes on the obstacle surface. When node $i$ locates at $\boldsymbol{\eta}_i(k-1)$, the length of the vertical line between the node and the obstacle $O_1(k)$ is less than the safe distance $\delta_3$, the virtual node $\boldsymbol{\eta}'_{1,1}(k-1)$ occurs. When node $i$ moves to $\boldsymbol{\eta}_i(k)$, the distance between the node and $O_1(k)$, $O_2(k)$ are both less than $\delta_3$. Then $\boldsymbol{\eta}'_{1,1}(k)$, $\boldsymbol{\eta}'_{2,1}(k)$ and $\boldsymbol{\eta}'_{2,2}(k)$ occurs. When the distance is more than $\delta_3$, the virtual nodes disappear. The shape and location of the obstacles is changeable all the time, which does not affect the avoidance process. One obstacle may generate more than one virtual node, such as $\boldsymbol{\eta}'_{2,1}(k)$ and $\boldsymbol{\eta}'_{2,2}(k)$ in Fig. 1.

If the virtual nodes are considered as a part of neighbor list, then $U(\boldsymbol{\eta}(k))$ in the obstacle environment is

$$U(\boldsymbol{\eta}(k)) = \sum_{i=1}^{n}\left( \sum_{j\in N(i,k)} \phi(\left\|\boldsymbol{\eta}_i(k) - \boldsymbol{\eta}_j(k)\right\|) + \sum_m \sum_l \phi'(\left\|\boldsymbol{\eta}_i(k) - \boldsymbol{\eta}'_{m,l}(k)\right\|) \right) \quad (20)$$

$\phi'()$ denotes the distance constraint between the obstacle and the nodes. Its partial derivative with respect to $\boldsymbol{\eta}_i(k)$ is

$$\frac{\partial U(\boldsymbol{\eta}(k))}{\partial \boldsymbol{\eta}_i(k)} = 2\sum_{j\in N(i)} \frac{\partial\phi\left(\left\|\boldsymbol{\eta}_j(k) - \boldsymbol{\eta}_i(k)\right\|\right)}{\partial \boldsymbol{\eta}_i(k)} + \sum_m \sum_l \frac{\partial\phi\left(\left\|\boldsymbol{\eta}_i(k) - \boldsymbol{\eta}'_{m,l}(k)\right\|\right)}{\partial \boldsymbol{\eta}_i(k)} \quad (21)$$

## IV. Tracking Emission Sources

Most existing works focused on point target tracking, which means the targets are assumed as discrete points in the field. But in some cases such as the battle field, heat plumes from wildfire, radioactive or chemical clouds may exist and we need to find the emission source of these hazardous substances. The substances emitted by the source diffuse around. We can approximate the source location through tracking these substances. It is somewhat more complex, as the substances do not have a well-defined location. We need to know both the extent and the density of the substance. Because the source may exist at the point where the substance density is maximal, it also may be desirable to increase the node density in areas of great concentration while sacrificing detailed information about areas of low concentration or areas outside the substance.

In section IIIA, we have defined the quality measure for point target tracking. Now it will be improved for emission source tracking. The basic difference between point target tracking and emission source tracking is the target state value while the former is the target location and the latter is the substance density in the field $\Omega$. At any point $\mathbf{q}$, the change trend of the substance density is decided by the environmental parameters such as wind direction, air flow, temperature, etc. For example, when tracking a chemical gas under no wind condition, the simplified Gauss model [12] is always adopted as the gas diffusion model

$$\frac{\partial\rho(\mathbf{q},t)}{\partial t} = \sigma\left( \frac{\partial^2\rho(\mathbf{q},t)}{\partial \mathbf{q}_x^2} + \frac{\partial^2\rho(\mathbf{q},t)}{\partial \mathbf{q}_y^2} \right) \quad (22)$$

where $\sigma$ is the diffusion parameter. Eq. (22) indicates that the time-varying rate of the gas density is the summation of diffusion rates along the $x$ and $y$ axis. Then we discretize (22) into the transition model form as (2)

$$\begin{aligned} &\rho(\mathbf{q}_x,\mathbf{q}_y,k+1) = \rho(\mathbf{q}_x,\mathbf{q}_y,k) + \\ &\sigma\left( \begin{aligned} &\frac{\rho(\mathbf{q}_x+\Delta\mathbf{q}_x,\mathbf{q}_y,k) + \rho(\mathbf{q}_x-\Delta\mathbf{q}_x,\mathbf{q}_y,k) - 2\rho(\mathbf{q}_x,\mathbf{q}_y,k)}{\Delta q_x^2} + \\ &\frac{\rho(\mathbf{q}_x,\mathbf{q}_y+\Delta\mathbf{q}_y,k) + \rho(\mathbf{q}_x,\mathbf{q}_y-\Delta\mathbf{q}_y,k) - 2\rho(\mathbf{q}_x,\mathbf{q}_y,k)}{\Delta \mathbf{q}_y^2} \end{aligned} \right) \end{aligned} \quad (23)$$

where $\Delta\mathbf{q}_x$ and $\Delta\mathbf{q}_y$ is the unit axis space. The second term means the density change at $\mathbf{q}$ is brought by the density difference between $\mathbf{q}$ and its neighboring unit space. Eq. (23) can be used to predict the density of $\mathbf{q}$ at next time point.

Obviously, the more the density at some point is, the more its importance is and the more the tracking quality at this point should be. Each point is considered as a special point target. So the tracking quality measure of diffusing substance is set up as the product integral of the point tracking quality and the substance density over the region

$$Q(\boldsymbol{\eta}(k)) = \int_\Omega \rho(\mathbf{q},k) Q(\boldsymbol{\eta}(k),\mathbf{q}) d\mathbf{q} \quad (24)$$

$Q(\boldsymbol{\eta}(k),\mathbf{q})$ is the tracking quality measure at point $\mathbf{q}$ defined in Section IIIA. The optimization process of $Q(\boldsymbol{\eta}(k))$ will drive the nodes toward the region with higher density. Its derivative with respect to $\boldsymbol{\eta}_i(k)$ is given by

$$\frac{\partial Q(\boldsymbol{\eta}(k))}{\partial \boldsymbol{\eta}_i(k)} = \int_{\Omega} \rho(\mathbf{q},k) \frac{\partial Q(\boldsymbol{\eta}(k),\mathbf{q})}{\partial \boldsymbol{\eta}_i(k)} d\mathbf{q} \quad (25)$$

For $\mathbf{q} \notin s_i(k)$ ( $s_i(k)$ denotes the sensing region of node $i$), $\boldsymbol{\eta}_i(k)$ will not affect $Q(\boldsymbol{\eta}(k),\mathbf{q})$ . So Eq. (25) is simplified as

$$\int_{s_i(k)} \rho(\mathbf{q},k) \frac{\partial Q(\boldsymbol{\eta}(k),\mathbf{q})}{\partial \boldsymbol{\eta}_i(k)} d\mathbf{q} = \int_{s_i(k)} \hat{\rho}(\mathbf{q},k/k-1) \frac{\partial Q(\boldsymbol{\eta}(k),\mathbf{q})}{\partial \boldsymbol{\eta}_i(k)} d\mathbf{q} \quad (26)$$

The computation of $\frac{\partial Q(\boldsymbol{\eta}(k),\mathbf{q})}{\partial \boldsymbol{\eta}_i(k)}$ has been shown in (15). Here $\rho(\mathbf{q},k)$ is unknown, so we substitute it by $\hat{\rho}(\mathbf{q},k/k-1)$ .

## V. Simulation Studies

We have constructed a Matlab simulator to evaluate the algorithm performance. This simulator adopts the discrete event driven mechanism. The virtual clock drives the event execution in the simulation. At every time instant, the inner event scheduler chooses the earliest event in the candidate event queue, executes its handling function, generates new events with assigned time stamp and schedules these events in the event queue. So on this simulator, the packet exchange and transmission can be simulated closely to the distributed execution in reality.

### *A. Execution Snapshot*

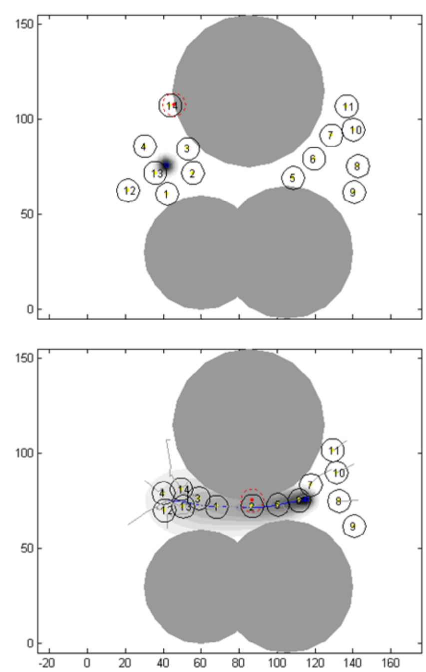

Fig.2. Execution snapshot of source tracking with obstacle existence

Fig.2 shows how the algorithm works. The communication radius is set to 30m and the sampling interval is 1s. The obstacles are made up of the big gray disks which locate within the field. The black line circles are the nodes and the red line circles are the location of the virtual nodes generated by the obstacles. The minimum safe distance is set to 3m. A gas tank car moves randomly and let out dangerous gas with a constant speed. The blue line is the car trajectory and the gray shadow represents the level of the gas density. The point with the highest density indicates the car location. The gray lines are the trajectories of the active nodes.

At every time only the nodes near the region of high density are active while the rest nodes maintain idle. The algorithm is fully distributed, so the sensing regions may happen to overlap each other. The car goes through the corridor between the obstacles. A few nodes locate at both ends of the corridor. With the car moving on, some nodes are drawn into the corridor to follow the car. They generate the virtual nodes based on their measurements. When the car moves near the other end, some nodes are activated by current active nodes and move toward the car to take over the tracking task. The network maintains the car visible all the time by tracking the gas region of high density.

### *B. Variance of Sensing Noise*

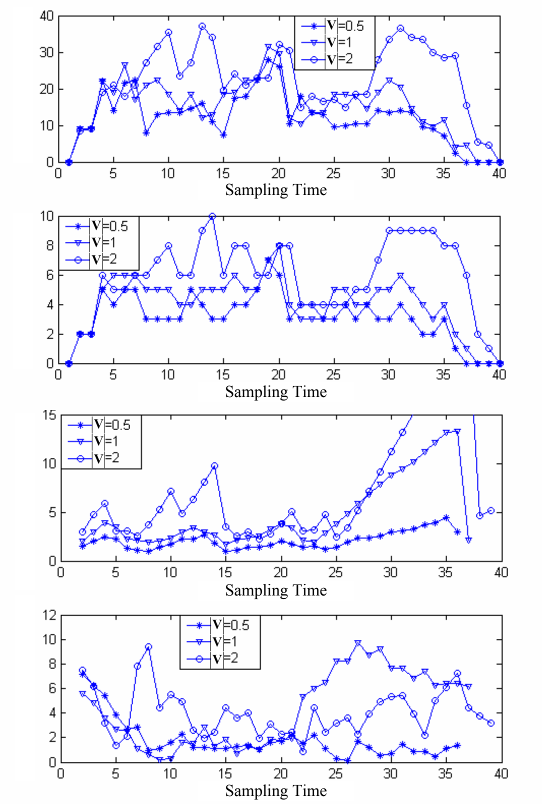


Fig. 3. Performance under different sensing noises

The target moves randomly in the field with a constant speed value 5m/s. The velocity constraint of each node is 3m/s and a acoustic amplitude sensor is fixed on each node. The sensor only detects the distance between the sensor and the target, so the measurement information is very limited. The energy efficient $\beta$ in Eq. (7) is set to 0.001.

Fig. 3 illustrates the relationship between energy conservation and the tracking quality when the variance $|\mathbf{v}|$ of sensing noise is different. We evaluate the algorithm performance from four aspects: Energy consumption is measured by the number and the moving distance of active nodes; the tracking quality is measured by the covariance determinant and the estimation error between the real location and the estimated location of the target. Obviously, it will cost more energy when using more uncertain sensors to track the target. But the estimation error is still larger because the energy conservation should also be considered.

### *C. Target-to-Node Speed Ratio*

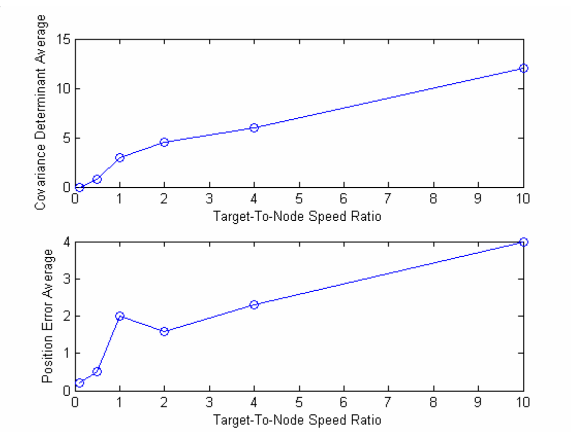


Fig.4. Performance under different target-to-node speed ratios

The algorithm performance is also related to factors such as the speed of the nodes, the target speed, the node density and so on. Fig. 4 shows the average of the position error over as well as the covariance determinant over the time sequence when the target –to-node ratio varies from 0.1 to 10. When the target is moving slowly, nodes are able to track accurately. If the target moves faster and faster, nodes cannot follow it any longer, which results the increase of the estimation uncertainty.

### *D. Node Density*

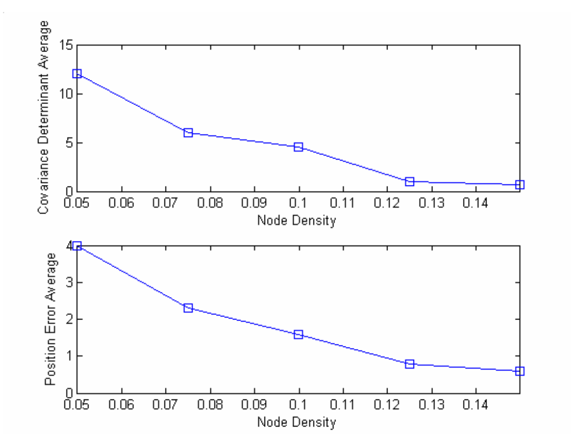


Fig.5. Performance under different node densities (nodes/ $m^2$ )

In Fig. 5, when the node density increases, the tracking quality is improved. This can be easily understood because the motion strategy of each node is decided by its negotiation with one-hop neighbors in a timely manner. Higher node density means more sensor data from neighbors is available for the node to make movement decision. But when the node density increases to a certain value, the uncertainty does not decrease greatly because the sensing coverage of nodes in the field almost reaches saturation and the existence of sensing noises prevent more increase of the tracking quality.

## VI. CONCLUSION

This paper proposed a distributed algorithm for a mobile sensor network to track motion of targets under sensing noses and existence of obstacles. The algorithm is suitable for tracking not only point targets but also diffusing targets. To incorporate sensing noises into the motion planning of nodes, we proposed to use the determinant of the covariance matrix of the target estimation as a new tracking quality measure. To cope with the obstacles in the environment, we transformed them to the virtual neighbors of the nodes so that obstacle avoidance and tracking can be integrated into a multi-objective optimization process. In a word, we have set up a widely applicable framework for distributed tracking problem using mobile sensor networks. It can be extended to complicated situations such as noisy sensing, obstacle existence, emission sources and so on. Numerical simulations have been conducted which confirmed good performance of the proposed algorithm.

## REFERENCES

[1] A. Makerenko, E. Nettleton, B. Grocholsky, S. Sukkarieh, and H. Durrant-Whyte, "Building a decentralized active sensor network," Proceedings of IEEE International Conference on Advanced Robotics, pp. 332-337, 2003.

[2] T. H. Chung, V. Gupta, J. W. Burdick, and R. M. Murray, "On a decentralized active sensing strategy using mobile sensor platforms in a network," Proceedings of IEEE International Conference on Decision and Control, vol. 2, pp. 1914-1919, 2004.

[3] J. R. Spletzer and C. J. Taylor, "Dynamic sensor planning and control for optimally tracking targets," International Journal of Robotics Research, vol. 22, pp.7-20, 2003.

[4] B. Jung and G. Sukhatme, "Tracking targets using multiple robots: the effect of environment occlusion," Journal of Autonomous Robots, vol. 12, pp. 191-205, 2002.

[5] L. E. Parker and B. A. Emmons, "Cooperative multi-robot observation of multiple moving targets," Proceedings of IEEE International Conference on Robotics and Automation, vol. 3, pp. 2082-2089, 1997.

[6] B. Shucker and J. K. Bennett, "Target tracking with distributed robotic macrosensors," Proceedings of Military Communications Conference, 2005.

[7] Y. Zou and K. Chakrabarty, "Distributed mobility management for target tracking in mobile sensor networks," IEEE Transactions on Mobile Computing, vol. 8, pp. 872-887, August 2007.

[8] Y. Y. Li, H. Y. Zhang, Y. H. Liu, "Distributed target tracking with energy consideration using mobile sensor networks," Proceedings of IEEE International Conference on Intelligent Robots and Systems, 2008.

[9] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, Nonlinear Programming: Theory and Algorithms. 2nd edition. Wiley, 1993.

[10] T. M. Cover and J. A. Thomas, Elements of Information Theory. 2nd edition. John Wiley & Sons, 2001.

[11] M. Brookes, "Matrix Reference Manual," Online, http://www.ee.ic.ac.uk/hp/staff/dmb/matrix/calculus.html.

[12] X. Y. Xia, Y. Song, D. Q. Fang, "Natural gas diffusion model and diffusion computation in well Cai25 Bashan Group oil and gas reservoir," Science in China Press, vol. 3, 2001.