WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
*Wirel. Commun. Mob. Comput.* 2008; **8**:1023–1033
Published online 10 July 2008 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/wcm.660

# Target tracking based on a distributed particle filter in underwater sensor networks

Yan Huang[1,2], Wei Liang[1], Hai-bin Yu[1*,†] and Yang Xiao[3]

[1]*Key Laboratory of Industrial Informatics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, P.R. China*
[2]*Graduate School of the Chinese Academy of Sciences, Beijing 100049, P.R. China*
[3]*Department of Computer Science, University of Alabama, Tuscaloosa, AL 35487, U.S.A.*

## Summary

In this paper, based on a distributed particle filter, two tracking algorithms are proposed for tracking mobile targets in cluster-based underwater sensor networks (USNs). Both tracking algorithms run local particle filter sequentially at each cluster along target trajectories, but they adopt different methods of selecting measurements from sensor nodes to balance the information contribution against the cost. Performance metrics are proposed and discussed in terms of tracking performance, communication cost, energy cost, and tracking response time. Simulations are conducted to quantitatively compare the proposed algorithms as well as another tracking algorithm based on extended Kalman filter (EKF). Our results indicate that one tracking algorithm achieves higher tracking accuracy while the other achieves dramatic reduction of communication cost, energy cost, and tracking response time. Furthermore, performance of two tracking algorithms has been studied in terms of detection threshold and sensor density. Copyright © 2008 John Wiley & Sons, Ltd.

KEY WORDS: underwater sensor networks; distributed particle filter; target tracking; performance evaluation

## 1. Introduction

In underwater sensor networks (USNs) [1,2], sensor nodes can perform sensing jobs with limited capacity of data processing and communication *via* acoustic modems. USNs have many potential civil and military applications such as monitoring marine environment for scientific exploration, commercial exploitation, and coastline protection. One important application is of tracking mobile targets, that is, on-demand target location estimation with measurements from sensor nodes.

Major challenges in designing tracking algorithms in USNs include (1) high power consumption with limited and unreplenishable power resources, and (2) communication constraints [3] such as limited bandwidth capacity, large propagation delay, and low reliability. Due to these unique characteristics of USNs, distributive and collaborative processing of measurements from multiple sensor nodes is recommended for reliable target tracking in this paper. Furthermore, it is desirable to minimize the cost of information processing in target tracking.

*Correspondence to: Hai-bin Yu, Key Laboratory of Industrial Informatics, Shenyang Institute of Automation, Chinese Academy of Sciences, 114 Nanta Street, Dongling District, Shenyang 110016, P.R. China.
†E-mail: yhb@sia.cn

In this paper, we design two tracking algorithms based on a distributed particle filter to track mobile targets in USNs. Sensor nodes are organized in a cluster-based architecture. To distribute computation load and communication load over the entire USN, both tracking algorithms run local particle filter sequentially at each cluster along the target trajectory. To balance the information contribution against the cost, the first tracking algorithm makes use of measurements beyond an adjustable detection threshold. Furthermore, we add an adaptive sensor selection step in the second tracking algorithm to reduce communication load and computation load. Performance metrics are proposed and discussed in terms of tracking performance, communication cost, energy cost, and tracking response time. Simulations are conducted to quantitatively compare the proposed algorithms as well as another tracking algorithm based on extended Kalman filter (EKF). Furthermore, performance of two tracking algorithms has been studied in terms of detection threshold and sensor density.

The remainder of this paper is organized as follows. In Section 2, we summarize the related research work along with a comparison with our work presented in this paper. Network, target, and sensor models are introduced in Section 3. In Section 4, we present two tracking algorithms based on a distributed particle filter for USNs. Performance metrics are proposed and discussed in Section 5. Section 6 presents the simulation results and analysis. Finally, Section 7 concludes the paper and describes directions for future work.

## 2. Related Work

Target tracking in deployable autonomous distributed system (DADS) has been studied in References [4–7]. In Reference [4], Mori *et al*. described two possibilities of tracking algorithms: one of them is to localize a target position from independent time difference of arrival (TDOA) measurements, requiring tremendous communication; the other is Doppler tracking, which is vulnerable to the target maneuvering. Puranik and Jannett [5] investigated various data fusion algorithms based on Kalman filter for target tracking. In Reference [6], an existing target tracking approach, distributed predictive tracking [7], was augmented using both stand-alone fuzzy logic and fuzzy logic with reinforcement learning techniques, to predict locations of targets in a manner which reduces the number of communication messages and sensing operations; target locations were obtained by aggregating location information from three sensors, which takes triangulation as an example. Recently, Amit *et al*. [8] explored the problem of activating an optimal combination of sensors to track a target moving through a network of underwater sensors; a new myopic sensor scheduling method executed on a scheduler (e.g., an aircraft or a surface ship) was developed to minimize predicted approximate tracking error subject to constraints on sensor usage and sensor costs.

Particle filter [9], which uses a set of particles to effectively represent the prior and posterior likelihood of states, is applicable for target tracking. Several particle filter based tracking algorithms are proposed for terrestrial sensor networks [10–12]. In Reference [10], Coates proposed a distributive particle filter algorithm in which each sensor node maintains a separate particle filter; however, this algorithm suffers high overhead of communication and computation. Sheng *et al*. [11] proposed two distributed particle filters, named as DPF-I and DPF-II, in which, observations by the sensors are divided into a set of disjoint uncorrelated cliques, and local particle filter runs sequentially at each clique or in parallel. Ma and Ng [12] proposed a tracking algorithm which is very close to DPF-I except using an objective function as a mixture of gain and cost to select sensing node; however, metrics of gain and cost, which may take various forms depending on the application and assumptions, have not been detailed in this algorithm; furthermore, they integrated EKF as a core component to propagate particles to high likelihood area.

In this paper, we focus on target tracking based on a distributed particle filter in a two-dimensional USN, where power consumption and communication constraints are stricter. There is a trade-off between performance and cost. Therefore, it becomes critical to carefully select sensor nodes that participate in collaboration. To reduce computation load and communication load, we study different methods of selecting measurements from sensor nodes for distributed particle filter based tracking algorithms. Moreover, we propose a performance evaluation system for the design of tracking algorithms for USNs.

## 3. Models for Underwater Tracking

In this section, network, target, and sensor models are introduced for the design of tracking algorithms for USNs.



DOI: 10.1002/wcm

### 3.1. Network Model

A two-dimensional USN [13] consists of sensor nodes which are anchored to the bottom of the ocean/lake to perform collaborative monitoring tasks over a given region, as shown in Figure 1. Sensor nodes reside at known static positions along the ocean floor that are assumed to be uniformly distributed with a mean sensor density of $\rho$ per unit of area. They are organized in a cluster-based architecture and each cluster has a cluster head (CH) node. Since using long fiber lines to connect the sensor nodes to CH nodes makes it vulnerable to trawling and dredging activities, and sometimes it is difficult to install wires, sensor nodes communicate with CH nodes by means of wireless acoustic links. CH nodes collect and integrate monitored data from sensor nodes and relay data from the ocean bottom network to a surface station, which is able to communicate with an onshore sink and/or to a surface sink.

### 3.2. Target Model

Assume that there is only one target moving within a plane at a known depth, according to the standard second-order model, as shown in Figure 2. It is assumed that the ocean floor is flat, and $h$ is the vertical distance between the ocean floor and the plane within which the target is moving.

$$X_k = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} X_{k-1} + \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{bmatrix} W_{k-1} \quad (1)$$

where $X_k = [x(k), \dot{x}(k), y(k), \dot{y}(k)]^{\mathrm{T}}$ a state vector; $W_{k-1} = [w_x(k-1), w_y(k-1)]^{\mathrm{T}}$ a system noise; $(\dot{x}(k), \dot{y}(k))$ a velocity vector, $(x(k), y(k))$ the two-dimensional position of the target at time $k$; $w_x(k-1)$ and $w_y(k-1)$ are both zero-mean Gaussian variable; and $T$ is the sampling time.

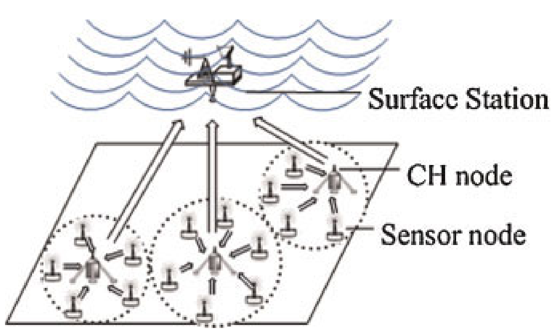


Fig. 1. Two-dimensional underwater sensor network.

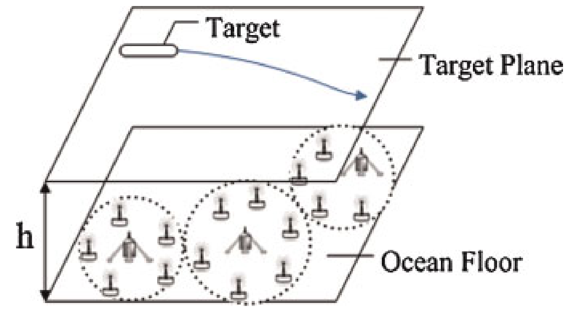


Fig. 2. Target moving plane.

### 3.3. Sensor Model

In this research, each sensor node has an acoustic sensor. A simplified model for an acoustic sensor consists of acoustic sound pressure measurement model (wide-band processing), base frequency measurement model (narrow-band processing), and acoustic spectrum pattern model (narrow-band processing) [4]. Like the target model, sensor model described in this section is only for the design of tracking algorithm, so that a simple sound pressure model is chosen. At time $k$, the received sound pressure of the sensor node $j$ is

$$z_j(k) = \frac{S(k)}{[x(k) - x_j]^2 + [y(k) - y_j]^2 + h^2} + \varepsilon_j(k) \quad (2)$$

where $S(k)$ is the target's source-level sound pressure; $(x(k), y(k))$ the two-dimensional position of the target at time $k$; $(x_j, y_j)$ the two-dimensional position of the sensor node $j$; and $\varepsilon_j(k)$ is an independent zero-mean random variable. It is assumed that all measurements from sensor nodes are independent.

## 4. Tracking Algorithms Based on DPF for USNs

In this section, we start with a brief review of a generic particle filter. Then we present two tracking algorithms based on a distributed particle filter for USNs, denoted as DPFTA-I and DPFTA-II, respectively.




DOI: 10.1002/wcm

## 4.1. Particle Filter

In target tracking, normally the system model and measurement model are adopted as follows:

$$x_k = F_k(x_{k-1}, v_{k-1}) \quad (3)$$

$$z_k = H_k(x_k, n_k) \quad (4)$$

where $\{x_k, k \in \mathbb{N}\}$ is a state vector; $\{z_k, k \in \mathbb{N}\}$ a measurement vector at time $k$; $\mathbb{N}$ the set of natural numbers; $F_k : \Re^{n_x} \times \Re^{n_v} \rightarrow \Re^{n_x}$ and $H_k : \Re^{n_x} \times \Re^{n_n} \rightarrow \Re^{n_z}$ are possibly nonlinear functions; $\{v_{k-1}, k \in \mathbb{N}\}$ an independent and identically distributed (i.i.d.) process noise; $\{n_k, k \in \mathbb{N}\}$ an i.i.d. measurement noise; and $n_x$, $n_y$, and $n_z$ are dimensions of $x_k$, $v_{k-1}$, and $n_k$, respectively.

Assume that the initial probability distribution function (PDF) $p(x_0|z_0) = p(x_0)$ of the state vector is known as the prior, where $z_0$ is the set of no measurements. Then, in principle, the PDF $p(x_k|z_{1:k})$ may be obtained, recursively, in two stages as follows, prediction in Equation (5) and update in Equation (6).

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})\,\mathrm{d}x_{k-1} \quad (5)$$

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})} \quad (6)$$

In Equation (5), the probabilistic model of the state evolution $p(x_k|x_{k-1})$ is defined by the system model (3) and the known statistics of $v_{k-1}$. In Equation (6), $p(z_k|z_{1:k-1}) = \int p(z_k|x_k)p(x_k|z_{1:k-1})\,dx_k$ is the normalizing constant, and the likelihood function $p(z_k|x_k)$ is defined by the measurement model (4) and the known statistics of $n_k$. The recurrence relations (5) and (6) form the basis for the optimal Bayesian solution. The sampling important resampling (SIR) filter, described in Table I—Algorithm 1, is a Monte Carlo method which can be applied to recursive Bayesian filtering problems [9]. $\{x_k^i,\ i = 1, \cdots, N_S\}$ is the set of particles with associated weights $\{\omega_k^i,\ i = 1, \cdots, N_S\}$ to represent the required posterior density function. $N_S$ is the number of particles. $K$ is the last sample time. $\{z_j(k),\ j = 1, \cdots, N(k)\}$ is the set of measurements at time $k$. $N(k)$ is the number of measurements at time $k$. $\mathrm{U}[a, b]$ is the uniform distribution on the interval $[a, b]$.

Table I. Algorithm 1: SIR particle filter.

**1.** Initialize $k = 0$
For $i = 1, \cdots, N_S$ Draw $x_0^i \sim p(x_0)$
**2.** For $k = 1, \cdots, K$
**2.1.** For $i = 1, \cdots, N_S$
– Draw $x_k^i \sim p(x_k|x_{k-1}^i)$
– Calculate $\omega_k^i = p(z_1(k), \cdots, z_{N(k)}(k)|x_k^i)$
– Normalize $\omega_k^i = \omega_k^i / \sum_{i=1}^{N_S} \omega_k^i$
**2.2.** $[\{x_k^j, \omega_k^j\}_{j=1}^{N_S}] = \mathrm{RESAMPLE}[\{x_k^i, \omega_k^i\}_{i=1}^{N_S}]$
– Initialize the cumulative distribution function (CDF): $c_0 = 0$
– For $i = 1, \cdots, N_S$ Construct CDF: $c_i = c_{i-1} + \omega_k^i$
– Start at the bottom of the CDF: $i = 1$
– Draw a starting point: $u_0 \sim \mathrm{U}[0, 1/N_S]$
– For $j = 1, \cdots, N_S$
· Move along the CDF: $u_j = u_0 + (j-1)/N_S$
· While $\mu_j > c_i$, $i = i + 1$
· Assign sample and weight: $x_k^j = x_k^i$, $\omega_k^j = 1/N_S$
**2.3.** Estimate the state and covariance:
$\hat{x}_k = \sum_{i=1}^{N_S} \omega_k^i x_k^i$, $C_k = \sum_{i=1}^{N_S} \omega_k^i (x_k^i - \hat{x}_k)(x_k^i - \hat{x}_k)^T$

## 4.2. Tracking Algorithm Based on Distributed Particle Filter

In a USN, two major challenges in designing tracking algorithms are power consumption and communication constraints such as limited bandwidth capacity, large propagation delay, and low reliability. Obviously, distributed implementation of tracking algorithm is a better choice. We propose two tracking algorithms based on the distributed particle filter, in order to distribute the computation load and communication load over the entire USN.

### *4.2.1. DPFTA-I*

Based on the hierarchical network architecture, a local SIR particle filter runs sequentially to update the posterior PDF of the target state. Only one cluster is active with SIR particle filter running at one time. However, not all sensor nodes in the current active cluster provide useful information that improves the estimate of target state. Some information may be even redundant. To balance the information contribution against the cost, sensor nodes whose measurements are beyond an adjustable detection threshold $D_\mathrm{t}$, report measurements to the CH node. The CH node calculates the posterior PDF of the target state from these measurements. If the CH node does not receive any measurements, the estimate of target state is replaced with the prediction. When the target moves out of the current cluster, the current CH node forwards its last estimation results (i.e., estimation of the state and covariance) to the next CH node which is the best candidate to track the



DOI: 10.1002/wcm

Table II. Algorithm2: DPFTA-I.

**1.** The initial CH node *CHa* does the following:
  **1.1.** For $i = 1, \cdots, N_S$ Draw $x_0^i \sim p(x_0)$
  **1.2.** For $k = 1, \cdots, K_a$
    **1.2.1.** For $i = 1, \cdots, N_S$ Draw $x_k^i \sim p(x_k|x_{k-1}^i)$
    **1.2.2.** If get measurements
      – For $i = 1, \cdots, N_S$
        · Calculate $\omega_k^i = p(z_1(k), \cdots, z_{N(k)}(k)|x_k^i)$
        · Normalize $\omega_k^i = \omega_k^i / \sum_{i=1}^{N_S} \omega_k^i$
      – Resample using **2.2** of Algorithm 1
    **1.2.3.** Estimate the state and covariance:
      $\hat{x}_k = \sum_{i=1}^{N_S} \omega_k^i x_k^i$, $C_k = \sum_{i=1}^{N_S} \omega_k^i (x_k^i - \hat{x}_k)(x_k^i - \hat{x}_k)^T$
  **1.3.** Forward $\hat{x}_{K_a}$ and $diag(C_{K_a})$ to the next CH node *CHb*
**2.** The CH node *CHb* does the following:
  **2.1.** For $i = 1, \cdots, N_S$ Draw $x_0^i \sim \mathrm{N}(\hat{x}_{K_a}, diag(C_{K_a}))$
  **2.2.** For $k = 1, \cdots, K_b$ Run as **1.2**
  **2.3.** Forward $\hat{x}_{K_b}$ and $diag(C_{K_b})$ to the next CH node *CHc*
**3.** The CH node *CHc* repeats the steps done by the previous CH node

mobile target. Then the new CH node makes a new estimation based on the estimation results from the previous CH node as well as new measurements from the sensor nodes in its own cluster. This procedure is repeated continuously until the target moves out of the monitored region.

The detection threshold can be defined beforehand. Also, it can be changed on demand according to the performance of the tracking algorithm, by sending a command. We will study performance of the tracking algorithms in terms of detection threshold in Subsection 6.3. The pseudo-code of DPFTA-I is shown in Table II—Algorithm 2, where $K_a$ and $K_b$ are the last sample times in the clusters of CH node CH$a$ and CH node CH$b$, respectively; diag($B$) returns the main diagonal of the matrix $B$; $\mathrm{N}(u, \sigma^2)$ is the normal distribution with mean $u$ and variance $\sigma^2$.

### 4.2.2. DPFTA-II

To reduce the communication load and the computation load, we propose DPFTA-II, which is similar to DPFTA-I. An adaptive sensor selection step is added except for the first sample time of each active cluster. At time $k$ ($k \neq 1$), the CH node calculates the predicted position of the target, and selects a sensor node which is the nearest to the predicted position by sending a report command. Only the selected sensor node reports measurement to the CH node at this sample time. Obviously, the performance of DPFTA-II depends on the accuracy of the predicted position and the measurement noise level of the selected sensor node. We propose to set a minimum acceptable detection threshold $D_{\mathrm{m}}$ ($D_{\mathrm{m}} > D_{\mathrm{t}}$). If the measurement of the selected sensor node is not beyond $D_{\mathrm{m}}$ at time $k$, the selection step is stopped provisionally at time $k + 1$. The pseudo-code of DPFTA-II is shown in Table III—Algorithm 3, where *flag* indicates if the selection step will be stopped at next sample time.

Table III. Algorithm3: DPFTA-II.

**1.** The initial CH node *CHa* does the following:
  **1.1.** For $i = 1, \cdots, N_S$ Draw $x_0^i \sim p(x_0)$
  **1.2.** $k = 1$, $flag = 0$
    **1.2.1.** For $i = 1, \cdots, N_S$ Draw $x_k^i \sim p(x_k|x_{k-1}^i)$
    **1.2.2.** If get measurements
      – For $i = 1, \cdots, N_S$
        · Calculate $\omega_k^i = p(z_1(k), \cdots, z_{N(k)}(k)|x_k^i)$
        · Normalize $\omega_k^i = \omega_k^i / \sum_{i=1}^{N_S} \omega_k^i$
      – Resample using **2.2** of Algorithm 1
    **1.2.3.** Estimate the state and covariance:
      $\hat{x}_k = \sum_{i=1}^{N_S} \omega_k^i x_k^i$, $C_k = \sum_{i=1}^{N_S} \omega_k^i (x_k^i - \hat{x}_k)(x_k^i - \hat{x}_k)^T$
  **1.3.** For $k = 2, \cdots, K_a$
    **1.3.1.** For $i = 1, \cdots, N_S$ Draw $x_k^i \sim p(x_k|x_{k-1}^i)$
    **1.3.2.** If $flag = 0$
      – Calculate the predicted position of target $x_P = \sum_{i=1}^{N_S} \omega_k^i x_k^i$
      – Select a sensor node nearest the predicted position
      – Send a report command and get the measurement $z_S$
      – If $z_S < D_{\mathrm{m}}$, $flag = 1$
      – For $i = 1, \cdots, N_S$
        · Calculate $\omega_k^i = p(z_s|x_k^i)$
        · Normalize $\omega_k^i = \omega_k^i / \sum_{i=1}^{N_S} \omega_k^i$
      – Resample using **2.2** of Algorithm 1
    **1.3.3.** Else
      – $flag = 0$
      – Run as **1.2.2**
    **1.3.4.** Run as **1.2.3**
  **1.4.** Forward $\hat{x}_{K_a}$ and $diag(C_{K_a})$ to the next CH node *CHb*
**2.** The CH node *CHb* does the following:
  **2.1.** For $i = 1, \cdots, N_S$ Draw $x_0^i \sim \mathrm{N}(\hat{x}_{K_a}, diag(C_{K_a}))$
  **2.2.** $k = 1$ Run as **1.2**
  **2.3.** For $k = 2, \cdots, K_b$ Run as **1.3**
  **2.4.** Forward $\hat{x}_{K_b}$ and $diag(C_{K_b})$ to the next CH node *CHc*
**3.** The CH node *CHc* repeats the steps done by the previous CH node

## 5. Performance Evaluation System

The performance of tracking algorithm for USNs should be measured from multiple aspects such as tracking performance, communication cost, energy cost, tracking response time. An ideal tracking algorithm for USNs has high tracking performance, low communication cost, low energy cost, and short tracking response time. To improve the tracking performance, more measurements from sensor nodes should be fused, and this leads to higher communication cost and energy cost. Therefore, the tracking performance conflicts with the communication cost and the energy cost. The tracking response time, which decides whether the target information can be obtained in time, is concerned closely with the characteristics of underwater acoustic communication.



DOI: 10.1002/wcm

The design of tracking algorithm for USNs requests for a kind of integrated evaluation system. We choose to use four metrics: tracking performance, communication cost, energy cost, and tracking response time. The formalized description of each performance metric is proposed in this section.

### 5.1. Tracking Performance

To indicate the accuracy of tracking algorithms, we adopt root mean square error (RMSE), defined as follows, to measure the tracking performance:

$$E_P(k) = \sqrt{\sum_{i=1}^{M} \frac{\left[(x_{k,i} - \hat{x}_{k,i})^2 + (y_{k,i} - \hat{y}_{k,i})^2\right]}{M}} \quad (7)$$

where $M$ is the number of Monte Carlo simulations; $(x_{k,i}, y_{k,i})$ and $(\hat{x}_{k,i}, \hat{y}_{k,i})$ are the true and the estimated two-dimensional positions of the target at time $k$, respectively, in the simulation run $i$.

### 5.2. Communication Cost

Communication traffic, which is the total amount of data transmission, is used to evaluate the communication cost. Under DPFTA-I, measurements beyond $D_t$ need to be transmitted. Under DPFTA-II, a report command and a selected measurement need to be transmitted except for that at the first sample time of each active cluster as well as the sample time when the selection step is stopped. Both tracking algorithms need to forward the estimation results when the active cluster is changing. Let $s_m$, $s_e$ and $s_c$ be the sizes (in terms of number of bits) of data packets of the measurement, the estimation results, and the report command, respectively. Obviously, we have $s_e > s_m > s_c$.

For the sake of simplicity, we assume that communication is reliable. If communication links are unreliable, the total amount of data transmission is related to the communication protocol and will be larger. The communication protocol is beyond the scope of our research. We compare the lower limits of the communication traffics of two tracking algorithms.

Let the communication traffics required for two tracking algorithms be $CT_I$ and $CT_{II}$, respectively, as follows:

$$CT_I = \sum_{k=1}^{N_{TS}} CT_I(k) = \sum_{k=1}^{N_{TS}} (s_m N_I(k) + s_e e(k)) \quad (8)$$

$$CT_{II} = \sum_{k=1}^{N_{TS}} CT_{II}(k) = \sum_{k=1}^{N_{TS}} (s_m N_{II}(k) + s_c r(k) + s_e e(k)) \quad (9)$$

where $CT_I(k)$ and $CT_{II}(k)$ are, respectively, the communication traffics required for two tracking algorithms at time $k$; $N_{TS}$ is the number of time steps; $N_I(k)$ and $N_{II}(k)$ are the numbers of measurements in two tracking algorithms, respectively; $N_I(k)$ is the number of measurements beyond $D_t$; $r(k)$ shows whether a report command has been sent (if yes, it returns 1; other return zero); and $e(k)$ indicates if the CH node has forwarded the estimation results.

At the first sample time of each active cluster as well as the sample time when the selection step is stopped, let $r(k) = 0$ and $N_{II}(k) = N_I(k)$ so that we have $CT_I(k) = CT_{II}(k)$. Otherwise, we have $r(k) = 1$ and $N_{II}(k) = 1$. If $s_m N_I(k) > s_m + s_c$ holds, namely $N_I(k) \geq 2$, we have $CT_I(k) > CT_{II}(k)$. When the active cluster is changing, let $e(k) = 1$, otherwise let $e(k) = 0$.

The average communication cost required for the algorithm $i$ of $M$ simulation runs is

$$CT_i = \sum_{j=1}^{M} \frac{CT_{i,j}}{M} \quad (10)$$

where $CT_{i,j}$ is the communication traffic required for the algorithm $i$ in the simulation run $j$.

### 5.3. Energy Cost

Sensor nodes are normally composed of four basic units: a sensing unit, a processing unit, a communication unit, and a power unit. Among these units, communication and sensing consume most of the energy. Furthermore, the energy consumed in sensing is the same for both tracking algorithms. We choose to neglect the energy consumed in computing and sensing, so that the energy cost depends on energy consumed in data transmission.

To quantify the energy cost in data transmission, we adopt the energy dissipation model based on the underwater acoustic communication principle in Reference [14]. To transmit a $b$-bit packet from one node to another over a distance $d$, the energy consumption of the transmitter is

$$E_T(b, d) = bP_0A(d) \quad (11)$$



DOI: 10.1002/wcm

and to receive this packet, the energy consumption of the receiver is:

$$E_R(b) = bP_r \tag{12}$$

where $P_0$ is a power level needed at the input to the receiver; $P_r$ is a constant parameter depends on the receiver devices; $A(d)$ is the attenuation, which is given as:

$$A(d) = d^m a^d \tag{13}$$

where $m$ is the energy spreading factor (1 for cylindrical, 1.5 for practical, and 2 for spherical) and $a = 10^{\alpha(f)/10}$ is a frequency-dependent term obtained from the absorption coefficient $\alpha(f)$ in dB/km for $f$ in kHz, where, we have

$$\alpha(f) = \frac{0.11f^2}{(1+f^2)} + \frac{44f^2}{(4100+f^2)} + 2.75 \times 10^{-4} f^2 + 0.003 \tag{14}$$

Let the energy costs required for two tracking algorithms be $EC_I$ and $EC_{II}$, respectively, as follows:

$$EC_I = \sum_{k=1}^{N_{TS}} EC_I(k) = \sum_{k=1}^{N_{TS}} \left\{ s_m \left[ P_0 \sum_{i=1}^{N_I(k)} A(d_i) + P_r N_I(k) \right] + s_e e(k) [P_0 A(d_{CH}) + P_r] \right\} \tag{15}$$

$$EC_{II} = \sum_{k=1}^{N_{TS}} EC_{II}(k) = \sum_{k=1}^{N_{TS}} \left\{ s_m \left[ P_0 \sum_{i=1}^{N_{II}(k)} A(d_i) + P_r N_{II}(k) \right] + s_c r(k) [P_0 A(d_s) + P_r] + s_e e(k) [P_0 A(d_{CH}) + P_r] \right\} \tag{16}$$

where $EC_I(k)$ and $EC_{II}(k)$ are the energy costs required for two tracking algorithms at time $k$, respectively; $d_i$ the distance between the CH node and the sensor node $i$, which reports measurement to the CH node; $d_{CH}$ the distance between two CH nodes; and $d_s$ is the distance between the CH node and the selected sensor node.

The average energy cost required for the algorithm $i$ of $M$ simulation runs is

$$EC_i = \sum_{j=1}^{M} \frac{EC_{i,j}}{M} \tag{17}$$

where $EC_{i,j}$ is the energy cost required for the algorithm $i$ in the simulation run $j$.

### 5.4. Tracking Response Time

The tracking response time, including data transmission time and computation time, indicates how long a USN can obtain the target information such as position, velocity. It is assumed that two tracking algorithms use the same computation time. The tracking response time at time $k$ is determined by the data transmission time, including the delay in waiting for being transmitted, the delay in sending or receiving the entire length of the packet over the channel, the propagation delay, etc.

Limited bandwidth capacity and large propagation delay are the unique characteristics of underwater acoustic communication. Since the sizes of data packets are small in both tracking algorithms, the propagation delay is used to measure the tracking response time. At time $k$, the tracking response times of DPFTA-I and DPFTA-II are expressed as follows:

$$TRT_I(k) = \sum_{i=1}^{N_I(k)} \frac{d_i}{ve} + \frac{e(k)d_{CH}}{ve} \tag{18}$$

$$TRT_{II}(k) = \sum_{i=1}^{N_{II}(k)} \frac{d_i}{ve} + \frac{r(k)d_s}{ve} + \frac{e(k)d_{CH}}{ve} \tag{19}$$

where $ve$ is the underwater acoustic propagation speed in m/s, which can be modeled as follows:

$$ve = 1410 + 4.21t - 0.037t^2 + 1.1\text{sa} + 0.018\text{de} \tag{20}$$

where $t$ is the temperature of the water in degrees Celsius; $sa$ the salinity of the water; and $de$ is the depth in meters.



DOI: 10.1002/wcm

## 6. Simulations

In this section, we present simulations for two tracking algorithms using MATLAB to study the trade-off between performance and cost. Two tracking algorithms are also compared with each other as well as with another tracking algorithm based on EKF, denoted as DEKFTA. DEKFTA is similar to DPFTA-I except running local EKF, which has been used in the context of DADS [5]. Furthermore, the two proposed tracking algorithms are studied in terms of detection threshold and sensor density.

### 6.1. Simulation Setup

Figure 3 shows the platform of the simulation scene. Sensor nodes are deployed over a region of size $2 \times 2$ km$^2$, with a mean sensor density of $\rho = 5$ per km$^2$. The whole region is divided into four clusters. $\mathrm{CH}_1(0.5, 0.5)$, $\mathrm{CH}_2(1.5, 0.5)$, $\mathrm{CH}_3(0.5, 1.5)$, and $\mathrm{CH}_4(1.5, 1.5)$ are the CH nodes. $\mathrm{SS}(1, 1)$ is the surface station. $h = 0.1$ km is the vertical distance between the ocean floor and the plane within which the target moves.

The actual initial state of the target is $X_0 = [2, -0.06, 0.5, 0.06]^{\mathrm{T}}$. The system noise is a zero mean Gaussian white noise process with covariance matrix $Q = \mathrm{diag}([0.0002, 0.0002])$. The initial state vector is assumed to have a Gaussian distribution with known mean $\overline{X_0} = [2.004, -0.062, 0.505, 0.058]^{\mathrm{T}}$, and covariance matrix $M_0 = \mathrm{diag}([0.004, 0.002, 0.005, 0.002])$. $S$ is set to $5 \times 10^5$. $T = 1$ s is the sampling time. The measurement noise is a zero mean Gaussian white noise process with variance $R = 0.5^2$. $D_{\mathrm{t}}$ is set to 2 and $D_{\mathrm{m}}$ is set to 10. The sizes of data packets are $s_{\mathrm{m}} = 32$ bits, $s_{\mathrm{e}} = 96$ bits, and $s_{\mathrm{c}} = 8$ bits. For the sake of simplicity, let $P_0 = P_r = 1$ mJ/bit, $m = 1.5$, and $f = 15$ kHz. Let $\mathrm{sa} = 34.5$, $\mathrm{de} = 250$ m, and $t = 10°$C, so that we have $ve = 1490$ m/s. The number of particles which are used for both tracking algorithms is $N_{\mathrm{S}} = 2000$.

We repeated 100 simulation runs of two tracking algorithms. In each run, we use the same target trajectory, but with different sequences of system noise and measurement noise. However, the mean and variance of system noise and measurement noise are the same.

### 6.2. Simulation Results

Tracking results of a simulation run are shown in Figure 4; and RMSE of 100 runs are shown in Figure 5, where divergent filtering has been excluded. Divergent condition is

$$\sqrt{(x_k - \hat{x}_k)^2 + (y_k - \hat{y}_k)^2} > 0.15\,\mathrm{km} \tag{21}$$

where $(x_k, y_k)$ and $(\hat{x}_k, \hat{y}_k)$ are the true (actual) and the estimated two-dimensional positions of the target, respectively, at time $k$. The number of divergent filtering for DEKFTA is 10, but DPFTA-I and DPFTA-II have no problem of filtering divergence. Although DEKFTA utilizes the same measurements as DPFTA-I, EKF has flaws when it is applied to nonlinear systems.

Illustrated from Figures 4 and 5, DPFTA-I achieves higher tracking accuracy than DPFTA-II. Because DPFTA-I utilizes more measurements and meanwhile DPFTA-II makes use of measurement from only one

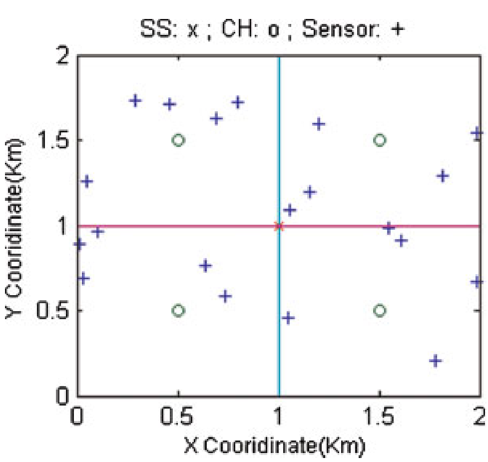


Fig. 3. Platform of simulation scene.

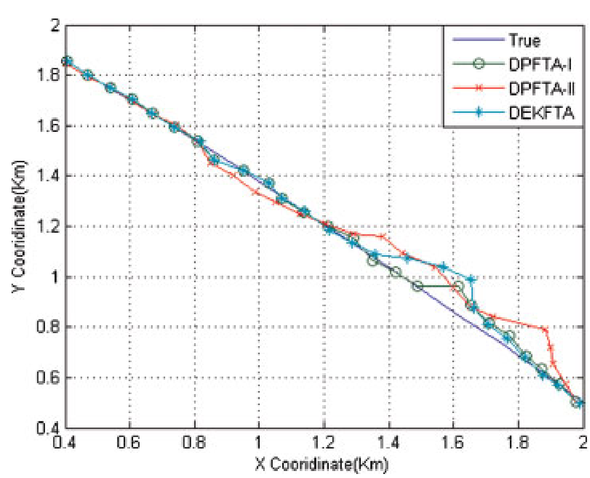


Fig. 4. Tracking results of a run.



DOI: 10.1002/wcm

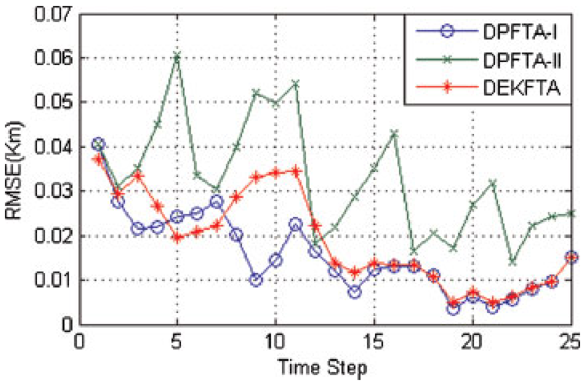


Fig. 5. RMSE of 100 runs.

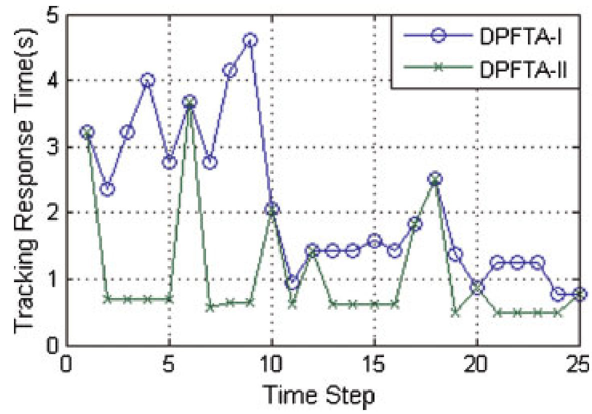


Fig. 6. Tracking response time of a run.

sensor node at most sample times. However, the average communication cost of DPFTA-I reaches 3197 bits when DPFTA-II reaches 1592 bits, only 50 per cent of DPFTA-I. The average energy cost of DPFTA-I reaches 8474 mJ when DPFTA-II reaches 3377 mJ, only 40 per cent of DPFTA-I. Furthermore, compared with DPFTA-II, the tracking response time of DPFTA-I at most of the sample times is longer, as shown in Figure 6. In this case, the USN is not able to take timely action, such as directing a tracker to intercept the target rapidly. In summary, DPFTA-II achieves dramatic reduction of communication cost, energy cost, and tracking response time at only small cost of tracking performance.

### 6.3. Simulation Analysis

In order to understand the performance of two tracking algorithms more comprehensively, we measure how the mean (Me), variance (Va) of RMSE, average communication cost (CT), and average energy cost (EC) vary with detection threshold $D_t$ and sensor density $\rho$ through simulations, as listed in Table IV.

Table IV shows that as $D_t$ increases, tracking performance expressed as the RMSE augments and tracking cost expressed as the average communication cost and energy cost decreases. Setting $D_t$ according to specific application requirements can result in different trade-offs between tracking performance and tracking cost. Furthermore, as $\rho$ increases, the tracking performance is improved and the tracking cost increases. However, there is a maximum $\rho$ beyond which using more sensors gains very little in the tracking performance. It is obvious that with the increment of $\rho$, the tracking performance of DPFTA-II is greatly improved with almost the same tracking cost. In conclusion, when $\rho$ is low, DPFTA-I is the better choice, otherwise DPFTA-II is a good substitute which can significantly reduce tracking cost at only a slight expense of tracking performance.

## 7. Conclusions

Target tracking is a representative application for USNs. We propose two tracking algorithms based on a distributed particle filter for two-dimensional

Table IV. Comparison under different $D_t$ and $\rho$.

| | $\rho = 5, D_t = 2$ | | $\rho = 5, D_t = 4$ | | $\rho = 8, D_t = 2$ | |
|---|---|---|---|---|---|---|
| | DPFTA-I | DPFTA-II | DPFTA-I | DPFTA-II | DPFTA-I | DPFTA-II |
| Me | 0.0157 | 0.0327 | 0.0237 | 0.0336 | 0.0101 | 0.0209 |
| Va | 0.0090 | 0.0128 | 0.0148 | 0.0138 | 0.0054 | 0.0102 |
| CT | 3197 | 1592 | 2054 | 1347 | 3990 | 1613 |
| EC | 8474 | 3377 | 5029 | 2573 | 11882 | 3374 |



DOI: 10.1002/wcm

USNs, denoted by DPFTA-I and DPFTA-II. For the design of tracking algorithms for USNs, a performance evaluation system is established. Simulations are conducted to quantitatively compare the proposed algorithms as well as another tracking algorithm based on EKF. The simulation results show that DPFTA-I is the better choice when sensor density is low; otherwise DPFTA-II is a good substitute which can significantly reduce communication cost, energy cost, and tracking response time at only a slight expense of tracking performance. Furthermore, setting detection threshold according to specific application requirements can result in different trade-offs between tracking performance and tracking cost.

In the next phase of our research, we will extend the tracking algorithms to track multiple targets and to be robust against transmission failures and out-of-sequence measurements.

## Acknowledgments

This work is supported by National Natural Science Foundation of China under grant numbers 60434030 and 60725312. Dr Yang Xiao's work is partially supported by the US National Science Foundation (NSF) under the grant number CNS-0716211. The authors thank the reviewers for their many helpful suggestions, which have greatly improved this paper.

## References

1. Cui J, Kong J, Gerla M, Zhou S. Challenges: building scalable and distributed underwater wireless sensor networks (UWSNs) for aquatic applications. *UCONN CSE Technical Report: UbiNet-TR05-02*, 2005.
2. Heidemann J, Li Y, Syed A, Wills J, Ye W. Underwater sensor networking: research challenges and potential applications. *USC/ISI Technical Report: ISI-TR-2005-603*, 2005.
3. Proakis J, Sozer E, Rice J, Stojanovic M. Shallow water acoustic networks. *IEEE Communications Magazine*, 2001; **39**(11): 114–119.
4. Doug H, Bruce D, Shozo M. Autonomous distributed systems: multiple-level distributed data fusion for future DADS using Bayesian network technology. *ADA399999* 2002.
5. Puranik S, Jannett TC. A comparison of the tracking performance of some distributed multisensor data fusion algorithms based on Kalman filter methods. In *Proceedings of the 35th Southeastern Symposium on System Theory*, 2003; pp. 455–459.
6. Yahya T. Applying fuzzy logic and reinforcement learning to track a mobile target using a wireless sensor network. *Ph.D. Dissertation*, University of Alabama in Huntsville, 2006.
7. Yang H, Sikdar B. A protocol for tracking mobile targets using sensor networks. In *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications*, 2003; pp. 71–81.
8. Chhetri AS, Morrell D, Papandreou-Suppappola A. Sensor resource allocation for tracking using outer approximation. *IEEE Signal Processing Letters* 2007; 213–216.
9. Arulampalam S, Maskell S, *et al*. A tutorial on particle filters for online nonlinear/non-Gaussian bayesian tracking. *IEEE Transactions on Signal Processing* 2002; **50**(2): 174–188.
10. Coates M. Distributed particle filters for sensor networks. In the *Third International Symposium on Information Processing in Sensor Networks*, 2004; pp. 99–107.
11. Sheng X, Hu Y, Ramanathan P. Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network. In *The Fourth International Symposium on Information Processing in Sensor Networks*, 2005; pp. 181–188.
12. Ma H, Ng BW-H. Collaborative data and information processing for target tracking in wireless sensor networks. In *IEEE International Conference on Industrial Informatics*, 2006; pp. 647–652.
13. Pompili D, Melodia T, Akyildiz IF. Deployment analysis in underwater acoustic wireless sensor networks. In *Proceedings of the 1st ACM International Workshop on Underwater Networks*, 2006; pp. 48–55.
14. Sozer E, Stojanovic M, Proakis J. Underwater acoustic networks. *IEEE Journal of Oceanic Engineering* 2000; **25**(1): 72–83.

## Authors' Biographies

**Yan Huang** is a Ph.D. student in Mechatronic Engineering of Shenyang Institute of Automation, Chinese Academy of Sciences. Her research areas are information processing in wireless sensor networks and underwater sensor networks.

**Wei Liang** received the Ph.D. in Mechatronic Engineering from Shenyang Institute of Automation, Chinese Academy of Sciences, in 2002. She is an associate professor with the Key Laboratory of Industrial Informatics, Shenyang Institute of Automation, Chinese Academy of Sciences. Her research areas are wireless sensor networks, underwater sensor networks, dynamic scheduling theory and system simulation.



DOI: 10.1002/wcm

**Hai-bin Yu** received the Ph.D. in Automation Control from Northeastern University, Shenyang, China. He is an ISA Senior Member and IFAC TC 1.5 Member. He is a professor with the Key Laboratory of Industrial Informatics, Shenyang Institute of Automation, Chinese Academy of Sciences. His research areas are wireless sensor networks, networked control systems and networked manufacturing.

**Yang Xiao** is currently a professor with the Department of Computer Science, University of Alabama. He is an IEEE Senior Member. His research areas are security, telemedicine, sensor networks, and wireless networks. He has published more than 200 papers in major journals (more than 60 in various IEEE journals/magazines), refereed conference proceedings, book chapters related to these research areas.



DOI: 10.1002/wcm