

Article

# Decentralized Cooperative TOA/AOA Target Tracking for Hierarchical Wireless Sensor Networks

Ying-Chih Chen and Chih-Yu Wen \*

Department of Electrical Engineering, Graduate Institute of Communication Engineering, National Chung Hsing University, Taichung 402, Taiwan; E-Mail: chenyingchr@gmail.com

\* Author to whom correspondence should be addressed; E-Mail: cwen@dragon.nchu.edu.tw; Tel.: +886-4-2285-1549 (ext. 811); Fax: +886-4-2285-1410.

Received: 6 July 2012; in revised form: 23 October 2012 / Accepted: 1 November 2012 /

Published: 8 November 2012

---

**Abstract:** This paper proposes a distributed method for cooperative target tracking in hierarchical wireless sensor networks. The concept of leader-based information processing is conducted to achieve object positioning, considering a cluster-based network topology. Random timers and local information are applied to adaptively select a sub-cluster for the localization task. The proposed energy-efficient tracking algorithm allows each sub-cluster member to locally estimate the target position with a Bayesian filtering framework and a neural networking model, and further performs estimation fusion in the leader node with the covariance intersection algorithm. This paper evaluates the merits and trade-offs of the protocol design towards developing more efficient and practical algorithms for object position estimation.

**Keywords:** wireless sensor networks; target tracking; Bayesian filtering; neural networking; covariance intersection

---

## 1. Introduction

Giving the limited power and processing capability in a sensor mote, a critical challenge of target tracking is how to acquire suitable data and perform information processing at the local level through cooperative communication and networking in the vicinity of the target. Thus, scalability and the need to conserve energy lead to the idea of hierarchically organizing the sensors, which can represent the target state and incorporate statistical models for the sensing schedule and target positioning. This paper aims

to develop a fully distributed method for cooperative target tracking in wireless sensor networks from two perspectives: (1) energy-balanced tracking and (2) improved estimation accuracy.

The first perspective is to build up an energy-balanced tracking network architecture. In this work, the concept of leader-based information processing is conducted to automatically achieve cooperative sensor scheduling with multiple tasking sensors in a cluster-based network topology based on sensor residual energy level, target information, and estimation quality. To avoid the ambiguity, the clusterhead and the cluster members refer to the original network topology, whereas the leader and the sub-cluster members refer to the sensor group for the tracking task. Random timers and local criteria are used to determine the tracking responsibility of the clusters. Afterwards, a sub-cluster of the corresponding cluster for the tracking task is formed by a leader, which can be a clusterhead or a cluster member in the original cluster-based network. The second perspective is to explore the behaviors/characteristics of a target such that supplementary information can be applied to improve estimation accuracy. Within the sub-cluster, the sensing nodes provide their measurements to the leader. Upon receiving the measurements, the leader fuses the local estimates from the sub-cluster members and reports it to the clusterhead. When the target moves out the region of the current active sub-cluster, the leader needs to trigger the leader handoff procedure (detailed in Section 3.4).

As shown in [1], compared with the dynamic clustering approach in a flat network topology, the static clustering approach incurs a larger location error since a clusterhead may not be a good local controller for estimating the location and reporting the event due to target movement. However, given a fixed hierarchical network topology, dynamic clustering approaches may not be feasible. Therefore, considering a cluster-based network topology, we introduce a distributed cooperative target tracking system, *Two-level Clustering Approach via Timer* (TCAT), which aims to improve the energy efficiency and provide good estimation accuracy. Here level-one clustering indicates the original network topology with control of clusterheads. Level-two clustering means the sub-cluster formation for the tracking task with control of tasking leaders. Therefore, the information flow goes through the sub-cluster members to the leader, and then to the clusterhead, and vice versa. Accordingly, the TCAT scheme performs target localization in four phases: (I) Tasking Leader Selection; (II) Choosing the Sub-Cluster Members; (III) Target Positioning; and (IV) Sub-Cluster Member Reselection and Leader Handoff.

In Phases I and II, random timers and local information are applied to adaptively select a tasking leader and sub-cluster members for the localization task. In Phase III, the Bayesian particle filter [2,3] is used to estimate the unknown target position from state equations. The objective is to find feasible position to minimize the error of the state vector. After obtaining the initial position estimate, the localization adjustment problem can be solved by applying a neural networking model, which focuses on improving positioning accuracy. Afterwards, the covariance intersection algorithm [4] is adopted to perform estimation fusion. In order to maintain tracking stability, Phase IV performs the leader handoff task.

The major contributions and key features of this paper are listed as follows. (1) We propose a novel cooperative positioning approach. (2) One of the main advantages of Bayesian framework is that the tasking sensor carries along a complete distribution of estimates of the target position. Thus, the distribution is inherently a measure of the accuracy of the positioning system. (3) Due to the characteristics of the cooperative information processing, the proposed estimation fusion approach owns

adaptive flexibility when dealing with uncertainty in position estimation. (4) In practice, two basic approaches can be applied to conserve power in a sensor network, either by the power management for sensor sleeping or by the design of low-power operation hardware. Our scheme is actually a distributed scheduling strategy for target tracking. The concept of “power management for sensor sleeping” is applied to select a subset of sensors for exploring detailed target information. The selected sub-cluster members (active nodes) are performing the tracking task and the inactive nodes can go to the idle/sleep mode. (5) We compare and contrast the existing tracking approaches with the proposed scheme. (6) We outline the technical foundations of the tracking techniques and present the tradeoffs in the algorithm design.

The rest of the paper is organized as follows. Section 2 briefly reviews the literatures on target tracking. Section 3 formulates the problem, derives a distributed solution, and describes a neural networking refinement model for target positioning. Section 4 presents an estimation-theoretic analysis of the proposed mechanism to assess the achievable estimation accuracy. Then, Section 5 considers the energy consumption of the proposed tracking scheme. In Section 6, the feasibility of the proposed scheme is examined via simulation. The performance comparison between the proposed approach and the scheme with a hierarchical network topology in [5] is presented. Section 7 makes a conclusion and shows future research directions.

## 2. Literature Review

There are five major categories for the target tracking solutions [1]: tree-based tracking, cluster-based tracking, prediction-based tracking, multicast message-based tracking, and hybrid tracking. Studies have shown that the cluster-based tracking algorithms have better network scalability and resource utilization compared with those in other categories. Prediction-based tracking rely on tree-based and cluster-based tracking in addition to the prediction method, but the tracking accuracy cannot be guaranteed. Multicast message-based tracking method depends on prediction, which is a multi-cast method in which message is delivered to a group of nodes that change with time according to estimated velocity of moving entity. Scheduling strategies vary in target tracking protocols and time synchronization may be needed to set the wake up and sleep timings of sensor nodes.

Since the proposed approaches fall into the category of cluster-based tracking, we focus on the research results of this category. The following subsections briefly describe the current literature of target tracking with respect to the number of tasking sensors.

### 2.1. Single Tasking Sensor

In the current literature, the general problem formation of target tracking is reformed to be a sensor selection problem with the uniform sampling interval and without incorporating the target dynamics, *i.e.*, the information-driven sensor query (IDSQ) approach [6] and the entropy approach based on sensor selection [7]. In contrast, the authors in [8,9] propose adaptive scheduling strategies to choose the single tasking sensor and determine the sampling interval simultaneously. In [8], the sensors are scheduled in two tracking modes: (1) the fast tracking approaching mode with the unsatisfactory predicted tracking accuracy; and (2) the tracking maintenance mode with the satisfactory predicted tracking accuracy.

The approach employs an Extended Kalman Filter (EKF) based estimation technique to predict the tracking accuracy, and adopts a linear energy model to predict the energy consumption. In [9], the proposed algorithm applies the interactive multiple model (IMM) filter to estimate and predict the target's dynamic state and select the tasking sensor node and sampling interval for each time step based on both the tracking accuracy and the energy cost. The simulation results show that the proposed approach outperforms the popular extended Kalman filter (EKF) based tracking scheme for maneuvering target in terms of tracking accuracy and energy efficiency. In [10], a small region is specified in order to select the single tasking sensor for achieving energy conservation. The distributed IMM filter is employed to estimate target position and velocity. A novel dynamic grouping idea is proposed to schedule next tasking node. However, the IMM filter has to face the problem of high complexity, especially with the operation of dynamic grouping.

In general, the localization problem can be solved by the joint time-of-arrival (TOA)/angle-of-arrival (AOA) positioning scheme using a single seed (*i.e.*, a sensor node with a known position). However, in the case of poor observations, more TOA/AOA measurements (*i.e.*, multiple tasking sensors) may be applied to complement the measurements of the environment.

## 2.2. Multiple Tasking Sensors

For the purpose of increasing estimation accuracy and reliability, multiple tasking sensors may be scheduled to track the target with detection uncertainties. In [5], target localization strategies based on a communication protocol between the clusterhead and cluster members are presented. In these approaches, a subset of sensor nodes is selected for querying detailed target information. Although energy and communication bandwidth are conserved in a certain amount, the processing burden all falls on the clusterhead, which may drain its power quickly. Suganya, *et al.* [11] focuses on tracking error and energy management involved in tracking the target. In this approach, the sensor nodes collectively monitor and track the movement of the target, which involves detecting, clustering and localization of target.

In [12], an information-driven approach in ad hoc sensor networks is proposed to determine the tasking sensors in a “sensor collaboration” by dynamically optimizing the information utility of data for a given cost of communication and computation. In [13], a distributed estimation method is proposed based on mobile agent (MA) computing paradigm and generic sequential Bayesian filtering for the target state estimation at each time step. Nonetheless, the MA migration planning problem needs to be handled in order to achieve the desired tracking accuracy. The tracking schemes in [14,15] combine the mechanisms of the tree-based and cluster-based schemes and propose information-based target tracking methods. However, the proposed sensor systems still have to deal with complexity issues. Authors in [16–21] propose multi-sensor scheduling schemes for maneuvering target tracking in sensor networks, while not considering the motion information of the target. Williams, *et al.* [22] presents an integrated approach to dynamically routing measurements and models in a sensor network and examines the problem of tracking objects within a region wherein the responsibility for combining measurements and updating a posterior state distribution is assigned to a single sensor at any given time step. The proposed approach is able to substantially reduce the communications cost incurred in tracking an object.

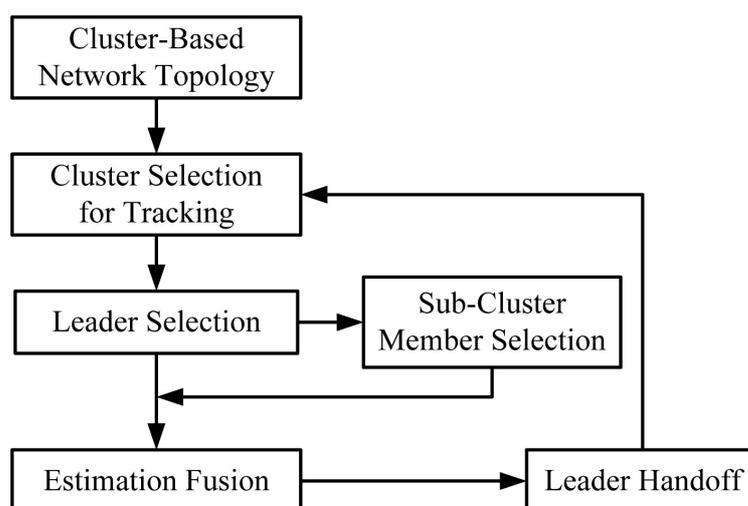
However, this strength comes along with the additional complexity of transmitting the representation of the state distribution between the tasking sensors. Comprehensive surveys of design challenges and recently proposed target tracking algorithms can be found in [1].

Note that most of these design approaches are dynamic clustering protocols in a flat network. In contrast, the method in [5] is built upon a static cluster-based network topology. Thus, reference [5] may provide a good way to benchmark the performance of the proposed tracking scheme.

### 3. Distributed Target Tracking Systems

In this section, we present the proposed distributed cooperative target tracking system: *Two-level Clustering Approach via Timer* (TCAT) in a cluster-based network topology, which organizes the tracking task in four phases: tasking leader selection, choosing the sub-cluster members, target positioning, and sub-cluster member reselection and leader handoff. Therefore, the proposed tracking approach organizes a sub-cluster for the tracking task, allows each sub-cluster member to locally compute the target position, and uses cooperation to obtain the fused estimate in the leader node. The local level estimate of a sub-cluster member and the global level estimate of a leader can be derived by a Bayesian and neural networking framework, and the covariance intersection algorithm, respectively. The main assumptions are: (i) all sensors are homogeneous; (ii) the sensors are in fixed and known location; (iii) a pre-specified sub-cluster size  $n$  is applied to perform cooperative target positioning with angle-of-arrival (AOA) information or hybrid time-of-arrival/angle-of-arrival (TOA/AOA) information, (iv) the target periodically broadcasts a message for measurement purpose. Note that these assumptions may be applied to healthcare scenarios or habitat monitoring to locate patients or animals. The distributed tracking architecture in a cluster-based network is depicted in Figure 1.

**Figure 1.** Illustration of block diagram for the TCAT method.

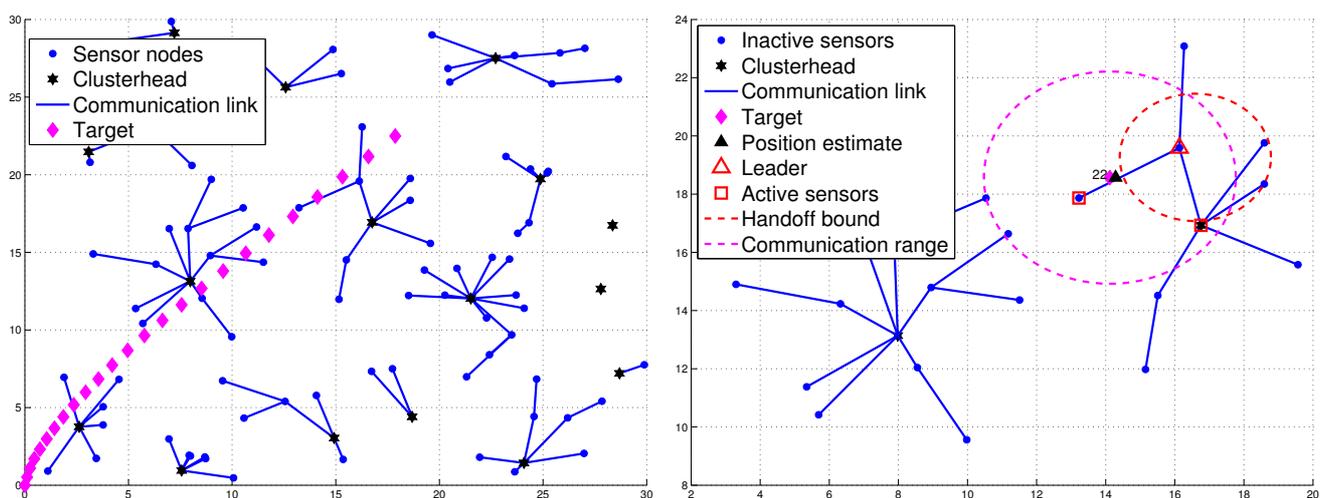


#### 3.1. Phase I: Tasking Leader Selection

When sensors are first deployed, they may apply the CAWT [23] to establish the cluster-based network architecture (Figure 2(left)). However, due to the target movement, the clusterhead may not be a proper

local controller in the neighborhood of the target. Thus, a cluster member may be a good control candidate and can be a leader for the tracking task. Denote a sensor with tracking responsibility as an active sensor. Otherwise, a sensor is marked as an inactive sensor. Suppose each sensor is an inactive sensor with the initial deployment. The tracking task is triggered when the target broadcasts a message of  $L_{id}$ , where  $L_{id}$  is a leader ID with an initial value zero, which is used to inform the active sensors to compete for being a leader of the tracking task. Thus, when sensor  $i$  receives the message sent from the target, it will broadcast a *Hello* message and become an active sensor, which allows itself to estimate how many neighboring active sensors it has. A *Hello* message consists of (1) the sensor ID of the sending sensor, (2) the leader ID of the sending sensor, and (3) the cluster ID of the sending sensor. Therefore, the sensors update their neighboring information (*i.e.*, a counter specifying how many neighboring active sensors it has detected) and decrease the random leader waiting time (LWT) through the received *Hello* message sent from neighboring active sensors.

**Figure 2.** The cluster-based network topology and target movement with 25 time steps (**left**); an example of leader and sub-cluster member selection (**right**).



Assume the initial value of the waiting time of sensor  $i$ ,  $LWT_i^{(0)}$ , is a sample from the distribution  $U(C, D')$ , where  $D' = C + D$ ,  $C$  and  $D$  are positive numbers, and  $U(\cdot, \cdot)$  is a uniform distribution. The update formula for the random LWT is given by

$$LWT_i^{(j+1)} = \alpha \cdot LWT_i^{(j)} \quad (1)$$

where  $LWT_i^{(j)}$  is the random LWT of sensor  $i$  at time step  $j$ , and  $0 < \alpha < 1$ . Note that the setting of random LWT may depend on sensor residual energy level, target information, and measurement quality (*e.g.*, the channel condition, the accuracy of positioning system). When the timer of sensor  $i$  expires, it then broadcasts a *Leader* message to claim that it is leader  $i$  (*e.g.*,  $L_{id} = i$ ) for the tracking task.

### 3.2. Phase II: Choosing the Sub-Cluster Members

Based on the claimed message sent from the leader and the cluster ID of the leader, the target will send a message to inform the active sensors about the corresponding cluster for the tracking task, which also

notifies the active sensors with the same cluster ID to be the candidate sub-cluster members associated with the leader. To choose the members associated with a leader, instead of directly selecting the active sensors from the leader, the sensor selection may be determined based on the reporting order of target position estimates from the neighboring active sensors of the leader. Accordingly, a candidate sub-cluster member, say sensor  $m$ , may decrease its LWT along with an extra backoff time  $BT_m$ , which is inversely proportional to the estimation accuracy, for reporting the estimate of target position. When the timer  $LWT_m$  expires, sensor  $m$  will deliver the tracking information to leader  $i$ . That is, based on the time stamps of the received estimates, the target tracking group is then automatically formed with the leader.

For those active sensors without receiving a *Leader* message, they transmit the estimated target position directly to the clusterhead and become supplementary sub-cluster members. This is attributed to the fact that the active sensors may not have direct communication with the leader. Hence, they may send their tracking estimates to the clusterhead for providing supplementary information on the tracking task. Therefore, when the number of sub-cluster members meets the desired number  $n$ , the leader will perform the CI model (as detailed in Section 3.3.3) to obtain a global target position estimate and send a *Position* message to the clusterhead, which also serves to specify the final sub-cluster members for cooperative target tracking. If the leader does not collect sufficient number of estimates (*i.e.*,  $|M| < n$ ,  $M$  is the index set of the sub-cluster), then the leader may send its fused estimate to the clusterhead and request the clusterhead to incorporate the supplementary estimates if possible. Afterwards, the clusterhead will perform the CI algorithm based on the received fused estimate and the supplementary estimates. Note that in order to spread the energy burden in the network, the cluster is responsible for informing the base station about the target tracking and positioning. Figure 2(right) shows an example of leader and sub-cluster member selection.

### 3.3. Phase III: Target Positioning

This phase presents a measurement mechanism to estimate the target position. The localization operation is performed in three phases: “initial geometrical positioning”, “position estimation refinement”, and “estimation fusion”. For each sub-cluster member, the Bayesian particle filter is used to obtain an initial position estimate. Next, the localization adjustment problem can be solved by applying a neural networking model to refine the sample area and then redoing the particle filter to improve the positioning accuracy. Finally, estimation fusion is performed by the leader node in order to obtain a global estimate.

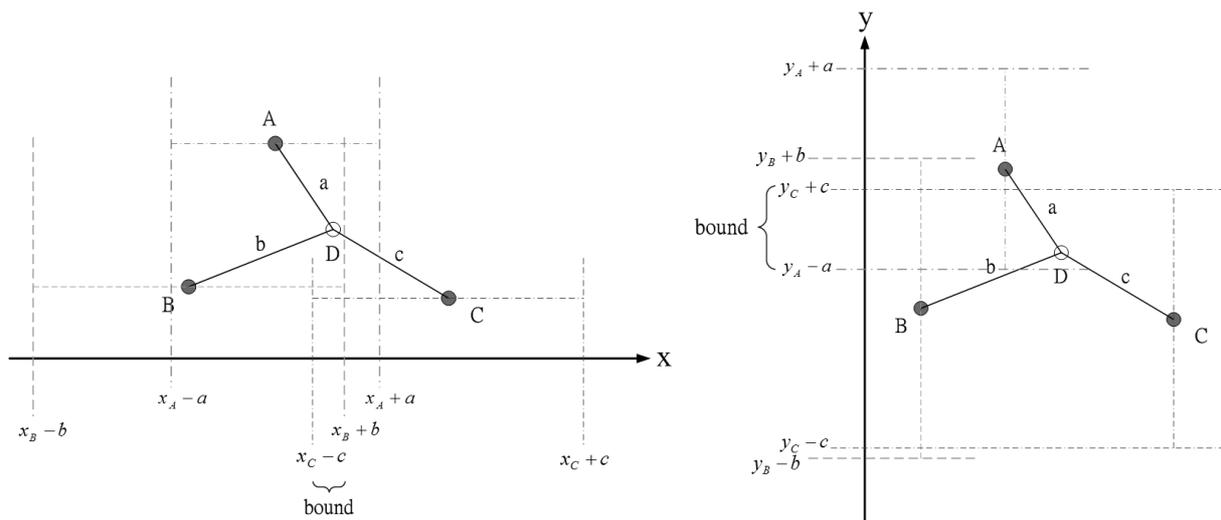
#### 3.3.1. Geometrical Positioning with Particle Filtering

The Bayesian particle filter [2] method may be preferred for object positioning because it is robust to noisy measurements, it allows for flexible information transmission, and it can be robust to lost or lossy data. Particle filter is an algorithm of estimation used to estimate the unknown target position from state equations. The objective is to find feasible position to make the error of state vector  $x$  minimum. The state vector is represented as a set of random samples updated and propagated with the algorithm. One of the main advantages of this approach is that the tasking sensor carries along a complete distribution of estimates of the target position. Therefore, the distribution is inherently a measure of the accuracy of the

positioning system. If a given task requires certain accuracy, it is possible to determine if that level of accuracy is currently available. Therefore, our approaches may be computationally affordable by sensor nodes.

The idea in [24], using known sensor positions and the bounding-box algorithm to extrapolate the unknown target position, inspires us to choose a proper prior density for generating initial samples. Figure 3 shows an example how the measurement information (e.g., distance information) can be used to obtain the  $x$  and  $y$  coordinate bounds of the unknown target. Therefore, the unknown target combines its bounds on the coordinates to form a bounding box, which provides a good set of initial samples for the particle filtering. In this work, each sub-cluster member uses Bayesian particle filter to estimate the unknown target position and performs target positioning with angle-of-arrival (AOA) information or hybrid time-of-arrival/angle-of-arrival (TOA/AOA) information. The particle filter method is shown in Table 1.

**Figure 3.** An example of obtaining the  $x$  and  $y$  coordinate bounds of the unknown sensor D by the distance and position information of the known sensors A, B, and C.



**Table 1.** The Particle Filtering Methodology.

1. **Initialization:** Generate a set of random samples  $x_k(i)$ ,  $i = 1, 2, \dots, N_{PF}$  from the prior density at time  $k = 0$ . Each sample of the state vector is a “particle”.
2. **Prediction:** Each random sample is passed through the state equation to obtain samples from the prior density at time  $k + 1$ . Thus

$$\hat{x}_{k+1}(i) = \Phi x_k(i) + \Gamma \lambda_k(i)$$

where  $\lambda_k(i)$  is a sample drawn from the probability density function of the system noise,  $\Phi$  is related to the mobility model, and  $\Gamma$  is an identity matrix.

**Table 1.** *Cont.*

- 
3. *Measurement Update:* The weights of the likelihood function  $p(z_{k+1}|\hat{x}_{k+1}(i))$  are updated for each sample in the random set  $i = 1, 2, \dots, N_{PF}$  and the normalized weights are

$$\xi_{k+1}(i) = \frac{p(z_{k+1}|\hat{x}_{k+1}(i))}{\sum_{j=1}^{N_{PF}} p(z_{k+1}|\hat{x}_{k+1}(j))}$$

for each sample.

4. *Re-sampling:* Take  $N_{PF}$  samples with replacement from the random sample set  $\hat{x}_{k+1}(i)$ ,  $i = 1, 2, \dots, N_{PF}$ , to generate the new sample set  $x_{k+1}(i)$ .
5. *Position:* The best single estimate of the position is the mean of  $x_{k+1}(i)$ ,  $\bar{x}_k$ .
- 

### 3.3.2. Position Estimation Refinement

Since the AOA measurement quality highly depends on the communication environment, this subsection presents estimation refinement criteria based on noisy AOA information, TOA information, and an angle-of-arrival neural networking (ANN) model. The purpose of the ANN model is to coordinate the initial target position estimate, the initial sample space, and the measurement information in a scenario with multiple tasking sensors such that effectively adjustment of angle information and a better sample area for particle filtering can be provided.

#### Angle-of-Arrival Neural Networking with CFBP

The feed-forward backpropagation (FFBP) and cascade-forward backpropagation (CFBP) are supervised learning algorithms for artificial neural networks which most commonly used for prediction, pattern recognition, and nonlinear function fitting [25]. Since the CFBP provides a better performance in terms of convergence time, optimum of network structure and recognition performance [26], the neural network with CFBP is applied for analyzing the performance of the TCAT.

Assume that the network under consideration has a general architecture with three layers of neurons. In our case, input and output layer neurons are linear, whereas neurons in the hidden layer are tan-sigmoidal. Let the vector pairs in  $\mathcal{T}$  be sample representation of the unknown function  $f: \mathcal{R}^n \rightarrow \mathcal{R}^p$ :  $T = \{(X_q, D_q)\}_{q=1}^Q$ , where  $n$  is the neuron index range in the input layer,  $p$  is the neuron index range in the output layer,  $X_q \in \mathcal{R}^n$ ,  $D_q \in \mathcal{R}^p$ ,  $Q$  is the number of training vector pairs, and  $q$  is the iteration index. Note that  $D_q$  is the desired vector response for the network input  $X_q$ . Thus, the mean square error of the entire training set is:  $\mathcal{E} = \frac{1}{Q} \sum_{q=1}^Q \mathcal{E}_q$ , where  $\mathcal{E}_q = \frac{1}{2} E_q^T E_q$ , and  $E_q$  is the instantaneous error of the training pair  $(X_q, D_q)$ .

Estimation Adjustment

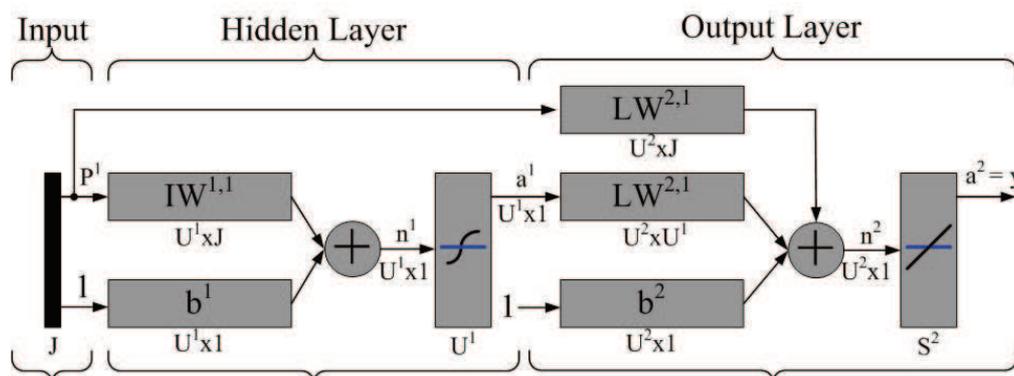
In order to adjust angle information, the three-layer perceptron neural network is considered. Based on the proposed network architecture, the goal of training is to maximize the correlation  $C$  between the signal of the hidden neuron and the residual output error [27]:

$$C = \sum_{j=1}^p \left| \sum_{q=1}^Q (S(z_h^q) - S_{av})(\delta_j^q - \Delta_j) \right| \tag{2}$$

where  $S(\cdot)$  is the signal function,  $S_{av} = \frac{1}{Q} \sum_{q=1}^Q S(z_h^q)$ ,  $\Delta_j = \frac{1}{Q} \sum_{q=1}^Q \delta_j^q$ ,  $z_h^q$  is the signal of the hidden neuron  $h$  in response to input pattern  $X_q$ ,  $\delta_j^q$  is the familiar scaled output error at neuron  $j$ ,  $\Delta_j$  is the average scaled error on the entire pattern set. Accordingly, for selecting the network parameters (weights and biases) that best approximate a given function, the backpropagation learning algorithm is considered to minimize the mean square error performance  $\mathcal{E}$ .

Figure 4 illustrates the perception network architecture. Note that  $J$  represents the number of input neurons, which may denote the number of received messages from neighboring sensors, the AOA measurement of the estimated target, the desired value of sub-cluster size  $n$ , and the minimum angle coverage area of a tasking sensor with a right-hand-side angle boundary and a left-hand-side angle boundary (two brown lines as shown in Figure 5(top)), where the initial sample area is located within the coverage area.  $U^1$  denotes the number of hidden neurons. In the output layer,  $U^2$  represents the number of neurons, which may denote the network approximation results. Moreover, let  $IW$  and  $LW$  be the input weight matrix and layer weight matrix for the hidden layer and the output layer, respectively. Let  $b^1$  and  $b^2$  be bias vectors for the hidden layer and the output layer, respectively. Established upon the developed neuron network model, the behavior of the TCAT scheme may be abstracted with sensible settings, which is further discussed in Section 6.

**Figure 4.** The three-layer perceptron network architecture for analyzing the performance of the TCAT.



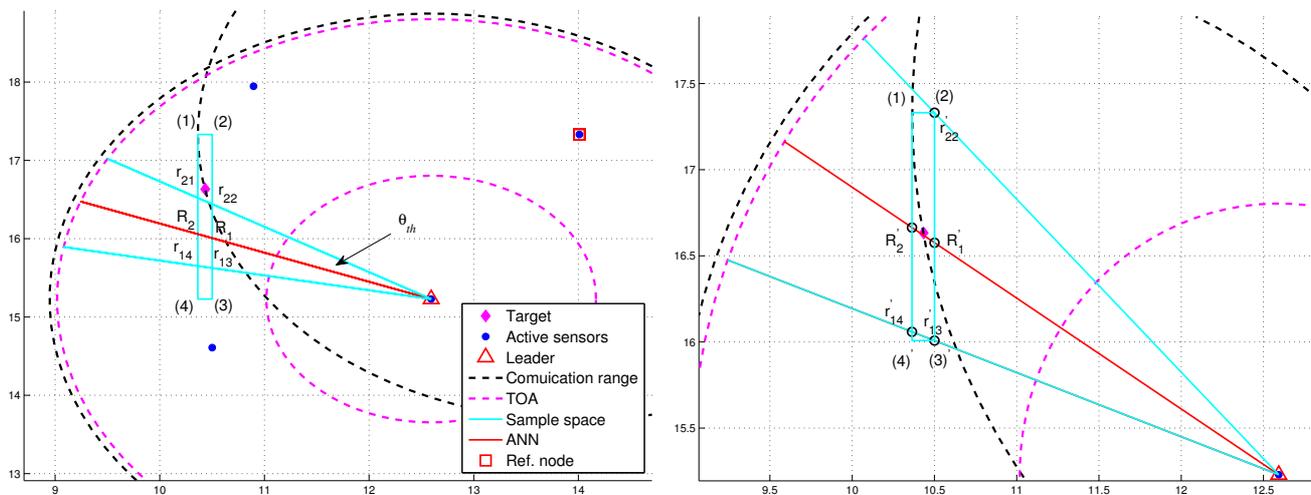


is the rectangle defined by blue lines and four corners: (1), (2), (3), and (4) ordered clockwise, which serves as a basis for estimation refinement. Observe that the target is located within the defined rectangle and the pink “TOA” circles represent the minimum and maximum range estimates.

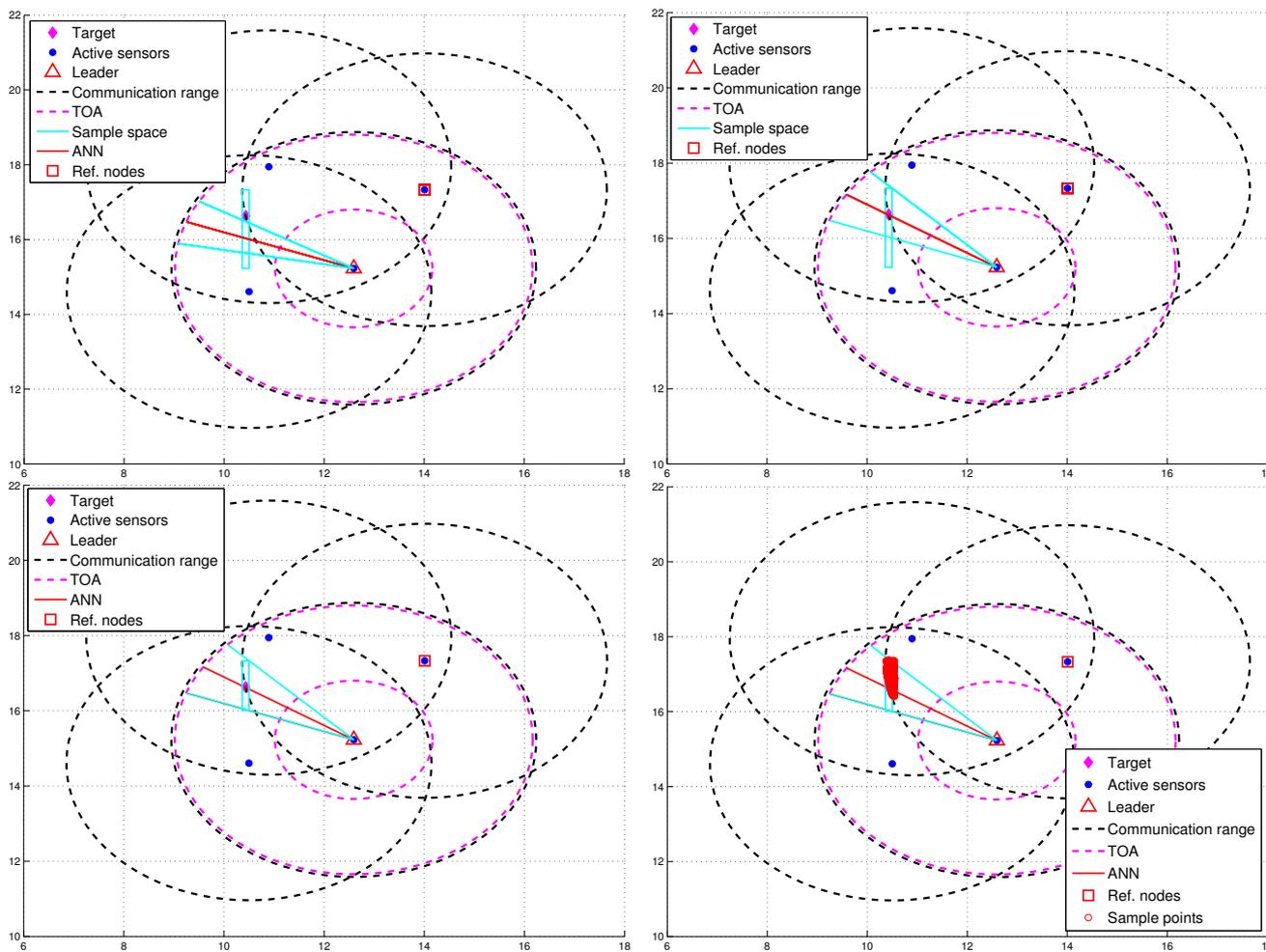
After importing the information of the defined rectangle and the tasking sub-cluster to the trained neural networking model, a reference angle information (RAI), (*i.e.*, the angle information label with a red line), is proposed to refine the initial sample area for the prior density in Step 1 of Table 1. Without loss of generality, a tasking leader is applied as an example to describe the refinement procedures. With the RAI produced by the neural networking model, a threshold  $\theta_{th}$  (*i.e.*, the angle between the red line and the blue line in Figure 5) is used to adjust the initial sample area. Given the information of transmission range and the position knowledge of sub-cluster members, the leader node can choose a sub-cluster member, which is an active sensor and has a minimum overlapping area between the communication coverage and the initial sample area, as a reference node to narrow down the sample area. This is attributed to the fact that the position information and the communication coverage of a reference node can be applied to help reshape the initial sample area for localizing the target. On the basis of the RAI (the red line), denote  $r_{1i}$  ( $r_{2j}$ ) as the intersection between the initial sample area and the right-hand-side angle boundary (the left-hand-side angle boundary), where  $i$  and  $j$  are related to the four corners. Let  $R_1$  and  $R_2$  be the intersections between the initial sample area and the RAI. For this typical scenario, the intersections among the initial sample area, the right-hand-side angle boundary, the left-hand-side angle boundary, and the RAI (*i.e.*,  $r_{11}$ ,  $r_{12}$ ,  $r_{23}$ ,  $r_{24}$ ,  $R_1$ , and  $R_2$ ) are mostly located within the transmission range of a reference node (Figure 5(bottom)). Referring to Figure 5(top), the intersections between the initial sample area and the communication coverage of the reference node can be regarded as new corners of the refined sample area (*i.e.*,  $(2)' \leftarrow (2)$ ,  $(3)' \leftarrow (3)$ , and  $(4)' \leftarrow (4)$ ) in Figure 5(bottom). Let  $A_{ref}$  be the communication coverage of the reference node. Thus, the final sample area is the intersection among  $A_{ref}$ , the refined sample area defined by blue lines and four corners: (1),  $(2)'$ ,  $(3)'$ , and  $(4)'$  ordered clockwise, and the RAI with a threshold value of  $\theta_{th}$ .

However, Figure 6(left) shows that the deviation of RAI from the target direction and improperly selected threshold values of  $\theta_{th}$  (e.g., too small values of  $\theta_{th}$ ) may lead to the exclusion of possible target locations during the refinement process, which results in an even worse estimate compared with the one without applying the neural networking model. Therefore, in order to avoid the estimation error caused by this scenario, the adjustment principle is to jointly consider the communication coverage of a reference node and the locations of the intersections (*i.e.*,  $P_{r_{1i}}$ ,  $P_{r_{2j}}$ , and  $P_{R_k}$ ). Accordingly, if none of the intersection locations is within the communication coverage of the reference node, the updated RAI may be rotated towards the reference node since the original RAI is highly deviated from the target direction. That is, referring to Figure 6(left) and Figure 6(right), if ( $P_{r_{1i}} \notin A_{ref}$ ,  $P_{r_{2j}} \notin A_{ref}$ , and  $P_{R_k} \notin A_{ref}$ ), the angle information may be updated by ( $r'_{13} \leftarrow R_1$ ,  $r'_{14} \leftarrow R_2$ , and  $r'_{22} \leftarrow r_{22}$ ). Hence, the left-hand-side angle boundary is replaced by the original RAI and the right-hand-side angle boundary is replaced by the line between the position of the leader node and the corner (2). Afterwards, the intersection of the refined sample area and  $A_{ref}$  forms the final sample area, which generates more effective samples for particle filtering and target positioning. Figure 7 shows an example of angle adjustment for position estimation refinement, which consists of four stages: the original RAI, the updated RAI, the refined sample area, and the final sample area for particle filtering.

**Figure 6.** The deviation of RAI from the target (**left**); the updated RAI (**right**).



**Figure 7.** The original RAI (**top left**); the updated RAI (**top right**); the refined sample area (**bottom left**); the final sample area for particle filtering (**bottom right**).



### 3.3.3. Covariance Intersection (CI)

For obtaining global estimates, we adopt covariance intersection to perform data fusion. The CI method of [4] provides the best estimate given the information available, which takes a convex combination of mean and covariance estimates that are represented in information space. Since these typical runs are independent, the general form is

$$P_{cc}^{-1} = \omega_1 P_{a_1 a_1}^{-1} + \cdots + \omega_n P_{a_n a_n}^{-1} \quad (3)$$

$$P_{cc}^{-1} c = \omega_1 P_{a_1 a_1}^{-1} a_1 + \cdots + \omega_n P_{a_n a_n}^{-1} a_n \quad (4)$$

where  $\sum_{i=1}^n \omega_i = 1$ ,  $n > 1$ ,  $a_i$  is the estimate of the mean from available information,  $P_{a_i a_i}$  is the estimate of the variance from available information,  $c$  is the new estimate of the mean, and  $P_{cc}$  is the new estimate of the variance.

### 3.3.4. Estimation Fusion

The distributed scheme is executed in two steps: (1) Group Estimation: local decisions are performed; (2) Estimation Fusion: a fusion rule is applied to combine the posterior density of the estimation from each member of the cooperative group in the leader sensor. Since the weight reflects the significance attached to the estimate, the next issue is to determine the weight  $\omega_i$  for each estimate and try to weight out faulty estimates. One strategy for choosing  $\omega_i$  is to use the utility measure. Since the utility of a sensor measurement is a function of the geometric location of the target, here we consider the Mahalanobis measure [28]. Hence, with respect to a neighboring system estimate characterized by the mean  $\mu_{m\ell}$  and covariance  $\Sigma$ , the utility function for sensor  $m$  is defined as the geometric measure

$$\mathcal{U}_{m\ell} = (\mu_{m0} - \mu_{m\ell})^T \Sigma^{-1} (\mu_{m0} - \mu_{m\ell}) \quad (5)$$

where  $\mu_{m0}$  is the local estimated target position of sensor  $m$  and  $\ell$  refers to a neighboring system estimate. In order to arrive at a consensus, the utility measure  $\mathcal{U}_{m\ell}$  can be shown to be  $\mathcal{U}_{m\ell} \leq 1$  [29]. Given the utility measure, two estimates can be allowed to be compared in a common framework and measure how much they differ  $|\mu_{m0} - \mu_{m\ell}|$ . Accordingly, the weights for the CI method in (3) are given by

$$\omega_\ell = \frac{\frac{1}{\mathcal{U}_{m\ell}}}{\sum_{k \in U_s} \frac{1}{\mathcal{U}_{mk}}} \quad (6)$$

where  $U_s$  is the index set of the neighboring estimates that pass the utility test. Otherwise,  $\omega_\ell$  is set to be zero. Notice that in this work  $m$  may refer to a tasking leader and  $\ell$  may refer to a sub-cluster member.

### 3.4. Phase IV: Sub-Cluster Member Reselection and Leader Handoff

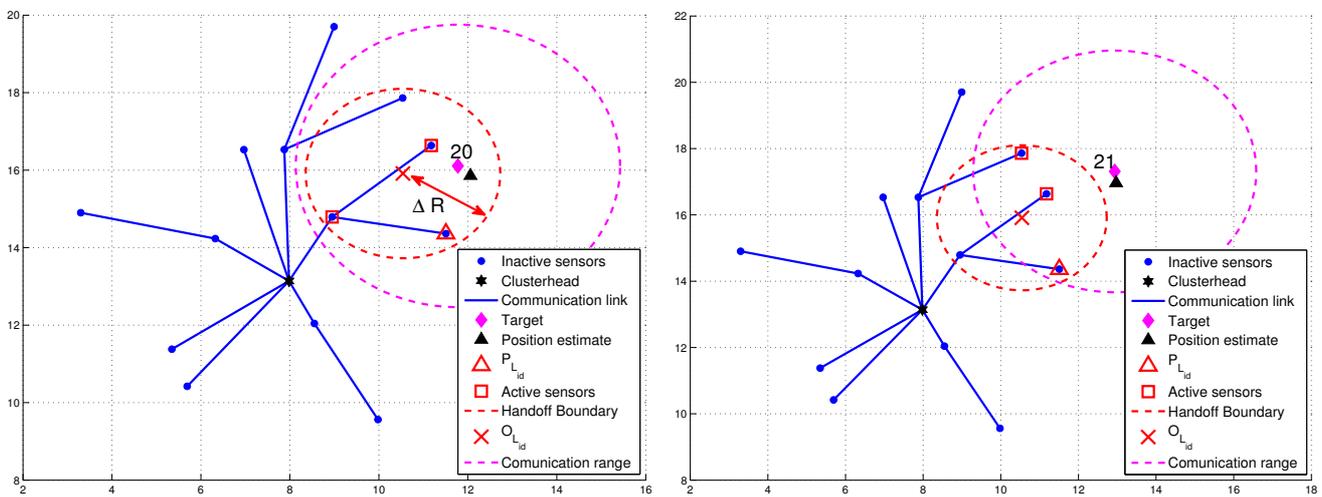
This phase performs the sub-cluster member reselection and leader handoff task, which aims to maintain tracking stability. The conditions for initiating the leader handoff procedure are:

- The distance between the reference location of the sub-cluster member  $P_i$  ( $\forall i \in M$ ) and the fused target position estimate  $f(j)$  exceeds the handoff threshold value at time step  $j$ . That is,  $d(P_i, f(j)) > \mathfrak{R}, \forall i \in M$ .

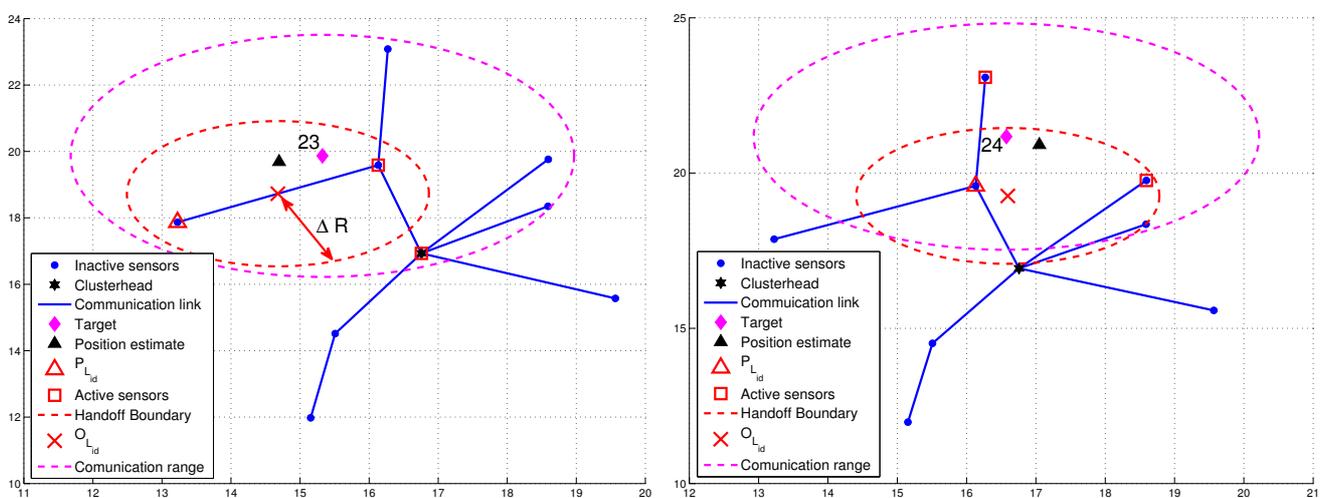
- Due to the movement of the target, the number of *Position* messages or expected active sensors are less than the desired value. That is,  $N_P < n$  or  $N_E < n$ .

For condition 1,  $\mathfrak{R} = \beta \cdot R$ , where  $R$  is the radio transmission range and  $0 < \beta < 1$ . For condition 2, a threshold value  $\Delta R$  with  $\Delta R = \delta \cdot R$  ( $0 < \delta < 1$ ) for dynamically measuring the number of valid sub-cluster members is applied. Denote  $P_{L_{id}}$  as the location of leader  $L_{id}$  and  $B_{L_{id}}$  as the leader handoff boundary (the red dashed line in Figures 8 and 9) of leader  $L_{id}$ , which is a circle of radius  $\Delta R$  centered at  $O_{L_{id}}$ . Note that  $O_{L_{id}}$  is the center of gravity of the bounding box, which is derived by the estimated target position, the positions of the sub-cluster members, and the bounding-box algorithm [24]. Thus,  $N_P$  is the number of estimates from the sub-cluster members and  $N_E$  can be computed as the number of active sensors which satisfy  $d(O_{L_{id}}, P_i) \leq \Delta R$ , where  $P_i$  is the location of active sensor  $i$ .

**Figure 8.** An example of adaptively updating the sub-cluster members with  $\Delta R$ .



**Figure 9.** An example of leader handoff procedure.



**Table 2.** Procedures of the TCAT model for target tracking.

- 
1. Target broadcasts a message with  $L_{id} = 0$ .
  2. Determine the active sensors  $I_A^{(j)}$  and the leader  $L_{id}$  at time step  $j$ .
    - (a)  $L_{id} = \operatorname{argmin}_i LWT_i^{(j)}, i \in I_A^{(j)}; M = L_{id}$ .
    - (b)  $BT_i = LWT_i + bf_i, i \in I_{CA}; I_{CA} = I_A^{(j)} \cap C_{L_{id}}$ .
  3. Find the sub-cluster members:
 

while( $I_{CA} \neq \emptyset$ )

    - (i) if( $(S = I_{N_{L_{id}}}) == \emptyset$ ),  $S = (I_{N_{L_{id}}} \cap I_{CA})^c$ .
    - (ii)  $\widehat{M} = \operatorname{argmin}_i BT_i^{(j)}, i \in S$ .
    - (iii) Send the position estimate to the leader or clusterhead.
    - (iv)  $M = M \cup \widehat{M}; S = (S \cap M)^c; I_{CA} = (I_{CA} \cap \widehat{M})^c$ .
    - (v) if( $|M| == n$ ), break.

end
  4. Estimation Fusion:
    - (a) Leader sends the global fused estimate  $f(j)$  to the clusterhead.
    - (b) The clusterhead disseminates the  $f(j)$  to the base station.
  5. Perform leader handoff (renewing sub-cluster members):
    - (a)  $N_P = |H_P|, H_P = \{i : d(T_j, i) \leq R, i \in M\}$ .
    - (b) if( $N_P < n$ ),  $j = j + 1$  and go to Step 1.
    - (c) if( $d(P_{L_{id}}, f(j)) > \mathfrak{R}$ )
      - $M = (M \cap L_{id})^c$ ,
      - $K = \{i : \operatorname{argmin}_i d(P_i, f(i)) \leq \mathfrak{R}, i \in C_{L_{id}}\}$ ,
      - $L_{id} = \operatorname{argmax}_k N_s(k), k \in K$ ,
      - if( $L_{id} == \emptyset$ ),  $j = j + 1$ ; go to Step 1.

end
    - (d)  $I_{CA} = (C_{L_{id}} \cap M)^c$ .
    - (e)  $N_E = |H_E|, H_E = \{i : d(O_{L_{id}}, P_i) \leq \Delta R, i \in M\}$ .
    - (f) while( $N_E < n$ )
      - (i)  $\widehat{M} = \operatorname{argmin}_i d(O_{L_{id}}, P_i) \leq R, i \in I_{CA}$ .
      - (ii)  $N_E = N_E + 1; M = M \cup \widehat{M}; I_{CA} = (I_{CA} \cap \widehat{M})^c$ .
      - (iii) if( $I_{CA} == \emptyset$ ), break.

end
    - (g) if( $N_E < n$ ),  $j = j + 1$ ; go to Step 1.
    - (h) Target broadcasts a message with  $L_{id}$  at time step  $j + 1$ .
    - (i) Go to Step 4.

---

Figure 8 presents an example of adaptively updating the sub-cluster members with  $\Delta R$ . Observe that at time step 20, two sub-cluster members are with tracking responsibility (Figure 8(left)). However, at time step 21, one sub-cluster member becomes an inactive sensor due to the target movement (Figure 8(right)). Thus, the leader will assign an active sensor which is located inside  $B_{L_{id}}$  to become

a new sub-cluster member. If any of the above conditions holds, the leader and target will sequentially broadcast a *Handoff* message with  $L_{id} = 0$  to trigger a leader reselection process as depicted in Phases I and II. Figure 9 shows an example of leader handoff procedure from time step 23 to time step 24. In Figure 9(left), a handoff procedure is triggered by condition 2. Afterwards, as shown in Figure 9(right), a new leader and its associated sub-cluster members are formed.

Due to the cluster-based network topology, the handoff schemes can be further divided into two categories: (1) intra-cluster leader handoff and (2) inter-cluster leader handoff. Since the clusterhead collects the supplementary estimates and receives the estimate from the leader, it may monitor the  $d(P_i, f(j))$ ,  $N_P$ , and  $N_E$ . If condition 1 holds, then an intra-cluster leader handoff is performed and the clusterhead may assign a cluster member to be a new leader. Otherwise, an inter-cluster leader handoff is triggered and the operations move to Phase I and II. The procedures of the TCAT model for cooperative target tracking are detailed in Table 2. Note that  $I_{N_i}$  is the index set of vicinal sensors of sensor  $i$ ,  $T_j$  is the true position of target at time step  $j$ , and  $C_i$  is the set of cluster members of sensor  $i$ .

#### 4. Analysis of Positioning Accuracy

Referring to [30], evaluating the computation process and the significance of approximate accuracy is an important step in deriving either exact or approximate solutions for the localization problem. This section presents an estimation-theoretic analysis of the proposed measurement mechanisms to assess the achievable localization accuracy with Cramer–Rao lower bound (CRLB) for joint TOA/AOA estimation.

The measurements at the reference sensor can be modeled as

$$\hat{\tau} = \tau + \delta_\tau \quad (7)$$

$$\hat{\phi} = \phi + \delta_\phi \quad (8)$$

where  $\tau$  is the true propagation time and  $\phi$  is the true angle information. Note that  $\delta_\tau$  and  $\delta_\phi$  are uncorrelated Gaussian noises with the distributions  $\delta_\tau \sim \mathcal{N}(0, \sigma_\tau^2)$  and  $\delta_\phi \sim \mathcal{N}(0, \sigma_\phi^2)$ . Assuming that the direct path exists between the sensor and the target, the estimated target position is given by

$$\hat{x} = x_s + v\hat{\tau} \cos(\hat{\phi}) = x_s + \hat{r} \cos(\hat{\phi}) \quad (9)$$

$$\hat{y} = y_s + v\hat{\tau} \sin(\hat{\phi}) = y_s + \hat{r} \sin(\hat{\phi}) \quad (10)$$

where  $\hat{r}$  is the distance measurement (*i.e.*,  $\hat{r} = v\hat{\tau} = r + v\delta_\tau$ ),  $(x_s, y_s)$  is the true position of the sensor and  $v$  is the speed of signal. Assuming  $\delta_\tau$  and  $\delta_\phi$  are sufficiently small, the variance of the position estimation  $\hat{p}$  is approximated by

$$\sigma_p^2 \approx v^2\sigma_\tau^2 + d^2\sigma_\phi^2 = \sigma_r^2 + d^2\sigma_\phi^2 \quad (11)$$

Given the above assumptions [31], the CRLB with a single sensor is derived as follows. The probability density function of  $\mathbf{g} = [\hat{r}, \hat{\phi}]$  is

$$f(\mathbf{g}; x, y) = \frac{1}{\sqrt{2\pi\sigma_r^2}} \cdot \exp\left[-\frac{1}{2\sigma_r^2} (\hat{r} - d)^2\right]$$

$$\frac{1}{\sqrt{2\pi\sigma_\phi^2}} \cdot \exp \left[ -\frac{1}{2\sigma_\phi^2} \left( \hat{\phi} - \arctan \left( \frac{y - y_s}{x - x_s} \right) \right)^2 \right] \quad (12)$$

Thus, the Fisher information matrix yields

$$I(\mathbf{x}^{(t)}) = \begin{bmatrix} \frac{\cos^2(\phi)}{\sigma_r^2} + \frac{\sin^2(\phi)}{d^2\sigma_\phi^2} & \frac{\sin(2\phi)}{2} \left[ \frac{1}{\sigma_r^2} - \frac{1}{d^2\sigma_\phi^2} \right] \\ \frac{\sin(2\phi)}{2} \left[ \frac{1}{\sigma_r^2} - \frac{1}{d^2\sigma_\phi^2} \right] & \frac{\sin^2(\phi)}{\sigma_r^2} + \frac{\cos^2(\phi)}{d^2\sigma_\phi^2} \end{bmatrix} \quad (13)$$

and the CRLB can then be written as

$$\text{Var}(\mathbf{x}^{(t)}) \geq I^{-1}(\mathbf{x}^{(t)}) = \begin{bmatrix} I'_{1,1} & I'_{1,2} \\ I'_{2,1} & I'_{2,2} \end{bmatrix} \quad (14)$$

Thus,

$$\text{Var}(\tilde{x}) \geq I'_{1,1}, \quad \text{Var}(\tilde{y}) \geq I'_{2,2} \quad (15)$$

Accordingly, the best possible fused estimation for target localization may be determined by applying Equation (15) and the CI method.

## 5. Analysis of Energy Consumption

This section considers the energy consumption of the proposed scheme in Section 3. It is often the case that inter-sensor communication costs are greater by orders of magnitude than local computation and sensing costs with respect to energy expenditures [22,32,33]. Therefore, the computational cost in a wireless sensor network is usually neglected compared to the communication cost. In addition, the problem of energy waste in idle listening mode can be solved by periodic listen and sleep. Moreover, based on the data sheet of CC2420 [34], the energy consumption of receive mode is about 59.1 mW, the energy consumption of transmit mode is about 52.2 mW, and the energy consumption of idle listening mode is about 1.2 mW. Accordingly, we focus on the comparison of power consumption for communication with the scheme of [5]. The total power requirements include both the power required to transmit messages and the power required to receive (or process) messages. Suppose that the energy needed to transmit for sensors with omnidirectional antennas is  $E_T$ , which depends on the transmitting range  $R$ , and the energy needed to receive is  $E_R$ .

When the target broadcasts a message with  $L_{id} = 0$ , its neighboring sensor, say sensor  $i$ , becomes an active sensor and broadcasts a *Hello* message with a random waiting time  $LWT_i^{(j)}$  for being a task leader at time step  $j$ . As the active sensor  $i$  claims to be a leader, the  $L_{id}$  is updated and broadcasted from target. As a result, the number of transmissions and receptions for tasking leader selection (Phase I) are

$$S_T^c(j) = 1 + N_t(j) \quad (16)$$

$$S_R^c(j) = \sum_{i \in (I_A^{(j)} \cup L_{id}^{(j)})} |I_{N_i}| + 2N_t(j) \quad (17)$$

where  $L_{id}^{(j)}$  is a leader ID at time step  $j$ ,  $I_{N_i}$  is the index set of vicinal sensors of sensor  $i$ , and  $N_t(j)$  is the number of transmissions of vicinal sensors of the target at time step  $j$ .

Afterward, the active sub-cluster members of the leader are selected according to the extra backoff time  $BT_m$ , which transmit the *Position* messages to the leader. Thus, the number of transmissions yields the sub-cluster size  $|M(j) \cap I_{N_{L_{id}^{(j)}}}|$  and the number of receptions is  $|I_{N_i}| (\forall i \in M(j) \cap I_{N_{L_{id}^{(j)}}})$ , where the  $M(j)$  is the set of sub-cluster members at time step  $j$ . Since the fused estimate  $f(j)$  is routed to the clusterhead in a multi-hop manner, the number of transmissions is  $N_i^H (\forall i \in (I_A^{(j)} \cap I_{N_{L_{id}^{(j)}}})^c)$ , where  $N_i^H$  is number of hops for sensor  $i$  to perform estimation reporting. Finally, the clusterhead disseminates  $f(j)$  to the base station. Furthermore, if the clusterhead receives a message from the leader for incorporating supplementary estimates, it may assign a desired number of supplementary members to be supplementary sub-cluster members for the tracking task. Therefore, the number of transmissions and receptions for selecting tasking members (Phase II) and delivering the position estimate (Phase III) are

$$S_T^p(j) = |M(j) \cap I_{N_{L_{id}^{(j)}}}| + \sum_{i \in (I_A^{(j)} \cap I_{N_{L_{id}^{(j)}}})^c} N_i^H + \sum_{i \in M'} N_i^H + 1 \quad (18)$$

$$S_R^p(j) = \sum_{i \in (M(j) \cap I_{N_{L_{id}^{(j)}}})} |I_{N_i}| + \left( \sum_{i \in (I_A^{(j)} \cap I_{N_{L_{id}^{(j)}}})^c} + \sum_{i \in M'} \right) \sum_{h=1}^{N_i^H} N_{I_{R_i}^{(h)}} \quad (19)$$

where  $M' = \{i : \operatorname{argmin}_i \sum N_i^H, i \in M(j) \cap (I_A^{(j)} \cap (I_{N_{L_{id}^{(j)}}} \cup L_{id}^{(j)}))^c\}$  and  $I_{R_i}^{(h)}$  is index set of relay ID for sensor  $i$  at hop  $h$ .

As characterized in Phase IV, since the clusterhead has the capability of updating the sub-cluster members, a cluster member, say sensor  $i$ , may become an active supplementary sub-cluster member or leader when receiving the message, which contains the IDs of new supplementary sub-cluster member or the leader (*i.e.*,  $(i \cap \overline{M}(j) \neq \emptyset)$  or leader ID), and then joint the tracking task. Note that  $\overline{M}(j)$  is the index set of new supplementary sub-cluster members. Therefore, we obtain

$$S_T^f(j) = \sum_{i \in M''} N_i^H + 1 \quad (20)$$

$$S_R^f(j) = N_{CH(j)} + \sum_{i \in M''} \sum_{h=1}^{N_i^H} N_{I_{R_i}^{(h)}} \quad (21)$$

where  $M'' = \{i : \operatorname{argmin}_i \sum N_i^H, i \in (\overline{M}(j) \cap I_{N_{CH(j)}})^c \cup (L_{id}^{(j+1)} \cap L_{id}^{(j)})^c\}$  and  $S_T^f(j)$  and  $S_R^f(j)$  are the number of transmissions and receptions for leader handoff, respectively. Nonetheless, when the leader ID is zero (*i.e.*,  $L_{id}^{(j+1)} = 0$ ), the sub-cluster members are assigned to be inactive sensors and the procedure of selecting a new sub-cluster will be triggered. Accordingly, the total energy consumption of transmission and reception for tracking target is  $E^{TCAT} = \sum_{j=1} (E_T \cdot (S_T^c(j) + S_T^p(j) + S_T^f(j)) + E_R \cdot (S_R^c(j) + S_R^p(j) + S_R^f(j)))$ .

## 6. Simulation

To evaluate the performance of the proposed approach, assume that the target moves within the  $x - y$  sensing field according to the standard second-order model [2]

$$X_k = \Phi X_{k-1} + \Gamma \mathbf{w}_k \quad (22)$$

over a four-dimensional state space, where  $X_k = (x, \dot{x}, y, \dot{y})_k^T$ ,  $\mathbf{w}_k = (w_x, w_y)_k^T$ , an uncorrelated Gaussian diffusion term describing the uncertainty,

$$\Phi = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \text{ and } \Gamma = \begin{pmatrix} 0.5 & 0 \\ 1 & 0 \\ 0 & 0.5 \\ 0 & 1 \end{pmatrix}$$

Here  $x$  and  $y$  denote the Cartesian coordinate of the target. The noisy measurement is given by

$$z_k = \tan^{-1}(y_k/x_k) + v_k \quad (23)$$

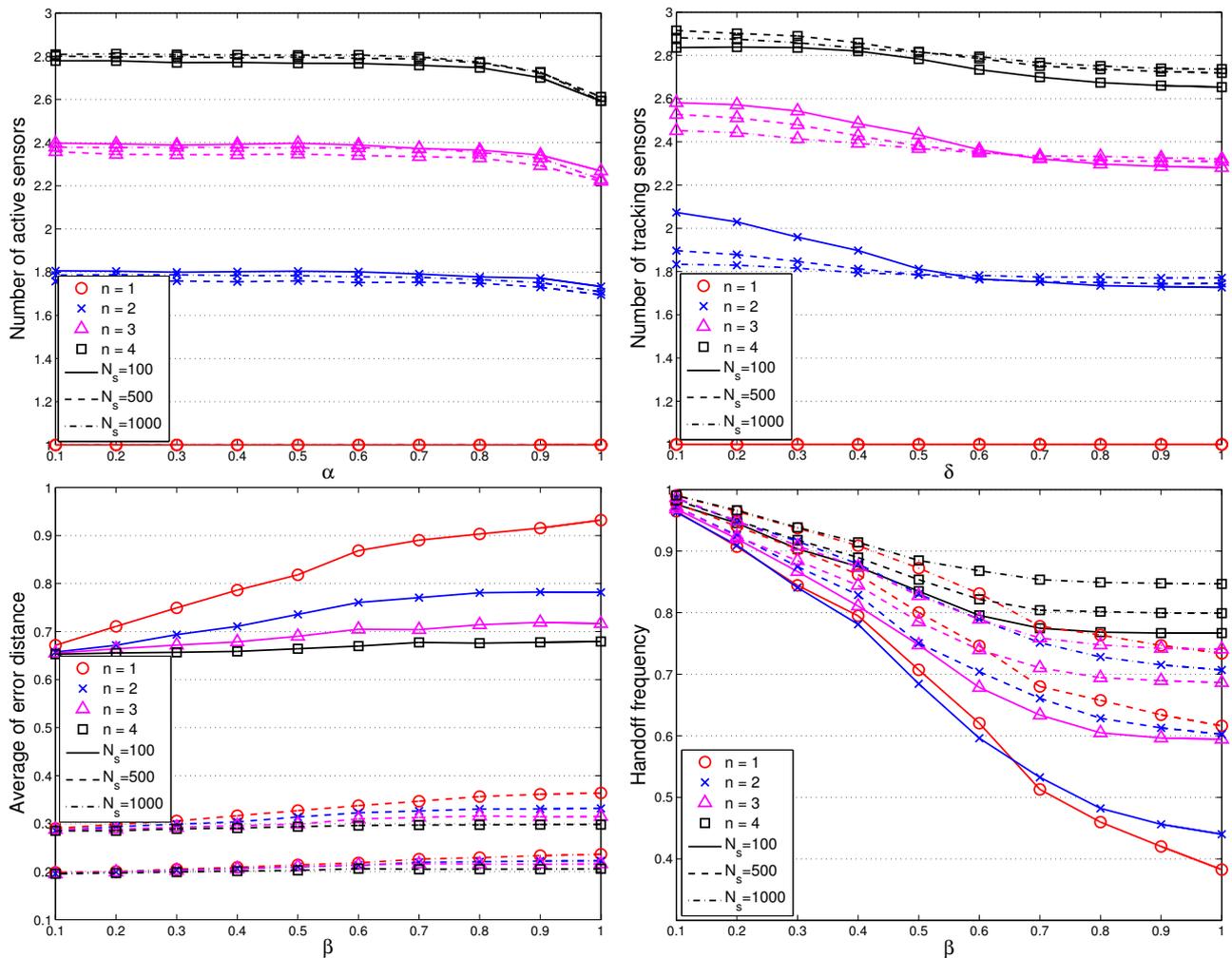
where the measurement noise,  $v_k$ , is a zero mean Gaussian white noise process with a finite variance  $\sigma_\theta^2$ . Before measurements are taken at  $k = 1$ , the initial state vector is assumed to be a Gaussian distribution with known mean  $\bar{x}_1$ , and covariance

$$M_1 = \begin{pmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \sigma_3^2 & 0 \\ 0 & 0 & 0 & \sigma_4^2 \end{pmatrix}$$

The target trajectory and measurements are generated based on Equations (22) and (23) with the parameter values: the covariance of the system noise,  $Q = qI_2$ , where  $I_2$  is the  $2 \times 2$  identity matrix,  $\sqrt{q} = 0.001$ . The initial state of the target is  $x_1 = (0.0, 0.1, 0.0, 0.05)^T$ . The prior distribution parameters are set to  $\bar{x}_1 = (0.0, 0.0, 0.4, 0.05)^T$  and  $\sigma_1 = 0.5$ ,  $\sigma_2 = 0.001$ ,  $\sigma_3 = 0.05$  and  $\sigma_4 = 0.01$ . The target position estimate is conducted with  $N_{PF} = 1000$  samples, the ANN model, and the CI method for 25 time steps.

Figure 10 depicts the system performance (e.g., the average positioning error and the leader handoff frequency) with various values of parameters ( $\alpha$ ,  $\beta$ ,  $\delta$ , sub-cluster size, and network density), for cooperative target tracking. Observe that for parameter  $\beta$ , there is a tradeoff between localization error and leader handoff frequency since a larger value of  $\beta$  (i.e., a larger handoff threshold value) may lead to a lower leader handoff frequency and may result in a lower positioning accuracy. Without loss of generality, we investigate the typical performance of the TCAT in a network with random uniform deployment of  $N_S$  sensors given  $\alpha = \beta = \delta = 0.5$ ,  $C = D = 1$ , and the standard deviation of angle information  $\sigma_\theta = 0.5$  radian. Note that the entire experiments are conducted in a square with side length  $L = 30$  unit length and transmission range  $R = L\sqrt{\log_{10}(L)/N_S}$  [35].

**Figure 10.** The target localization error with  $\beta = \delta = 0.5$  and varying the values of  $\alpha$  (**top left**); The target localization error with  $\alpha = \beta = 0.5$  and varying the values of  $\delta$  (**top right**); The target localization error with  $\alpha = \delta = 0.5$  and varying the values of  $\beta$  (**bottom left**); The leader handoff frequency with  $\alpha = \delta = 0.5$  and varying the values of  $\beta$  (**bottom right**).

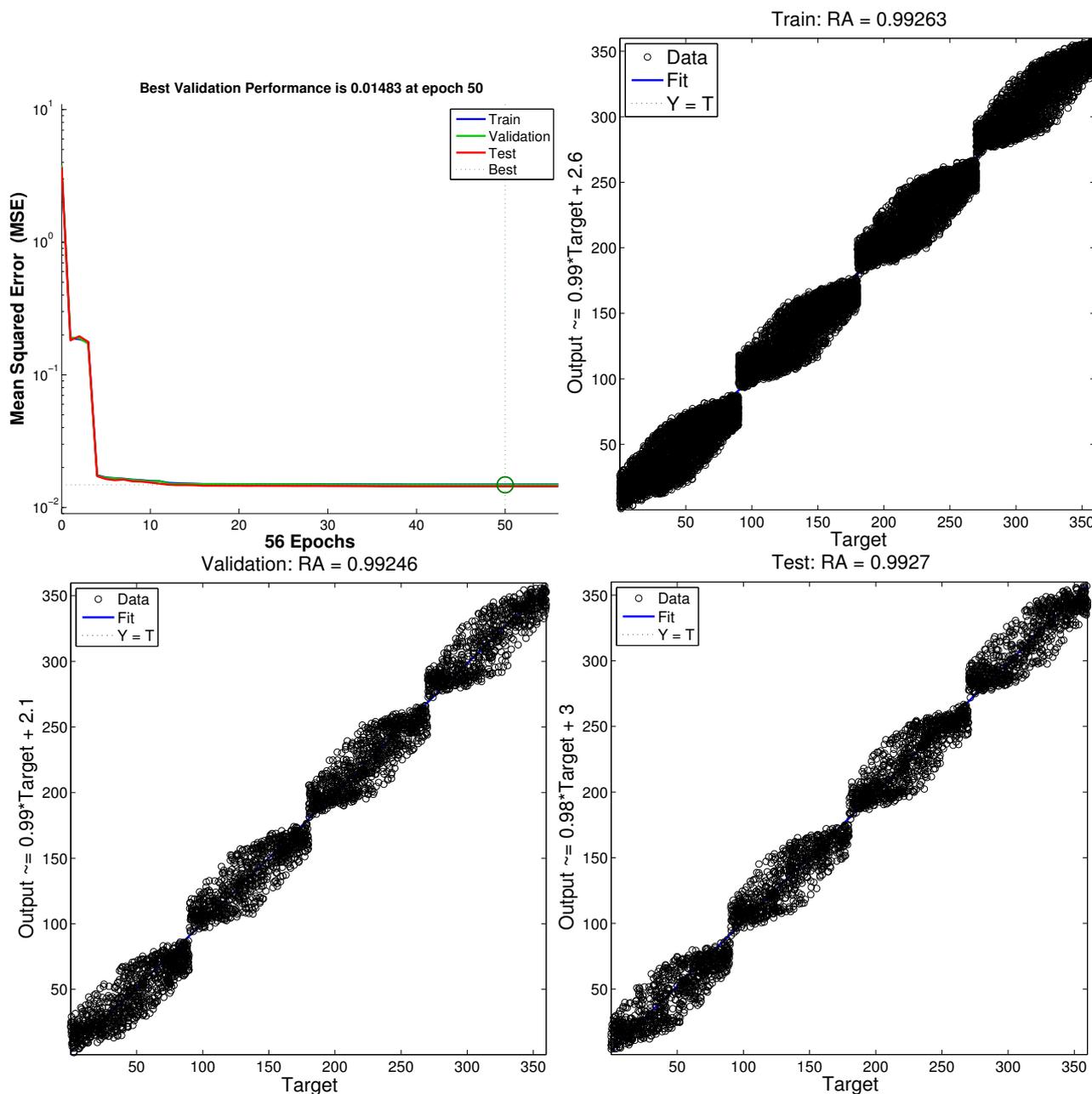


### 6.1. Performance of Neural Networking Model

For the CFBP model, Figure 11 depicts the learning and regression analysis of the network. Figure 11 (top left) shows that the network is learning since the mean squared error of the network is decreasing to a smaller value and the training of CFBP network is stopped before overfitting. The 22,804 input and target vectors are randomly divided into three sets. 15,962 vectors are used to train the network. Of these vectors, 3,421 are used to validate how well the network generalized. Finally, the last 3,421 vectors provide an independent test of network generalization to data that the network has never seen. Moreover, regression analysis is employed as post-training analysis between the network response and the corresponding targets and three parameters are returned to evaluate the performance. The first two parameters, slope and y-intercept of the best linear regression relate targets to network outputs. If the outputs exactly equal to targets, the slope and y-intercept would be 1 and 0, respectively. For the training case, slope = 0.99 and y-intercept = 2.6. For the validation case, slope = 0.99 and y-intercept = 2.1. For

the test case, slope = 0.98 and y-intercept = 3. The third parameter is correlation coefficient between the outputs and targets. When the correlation coefficient is equal to 1, then there is perfect correlation between targets and outputs. In this study, the correlation coefficients of the regression analysis is about  $RA = 0.99$  as shown in Figure 11, which therefore illustrates a good fit.

**Figure 11.** The performance analysis of the network (**top left**); the regression analysis between the network response and the corresponding targets: the training case (**top right**), the validation case (**bottom left**) and the test case (**bottom right**).



### 6.2. Target Localization Error

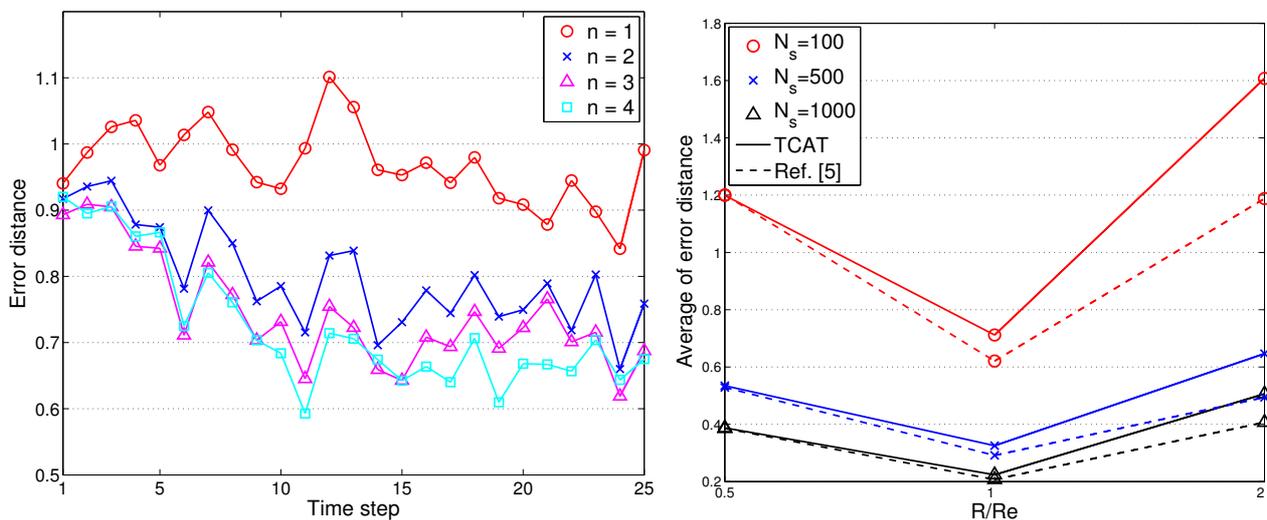
In order to further explore the effectiveness of the proposed refinement scheme, four different information measurement scenarios are considered: (1) AOA information only; (2) AOA with

neural network (ANN) refinement; (3) Joint TOA/AOA information only; (4) Joint TOA/AOA with ANN refinement.

### 6.2.1. AOA with/without ANN Refinement

To assess the tracking accuracy, the root mean square error is used for comparing the tracking accuracy of the distributed TCAT with that of [5]. Referring to the network topology and the target movement in Figure 2(left), we vary the number of sub-cluster members from 1 to 4. Figure 12(left) shows the accuracy of the position estimate. The performance improves along with the number of sub-cluster size  $n$ . However, the improvement is not significant (especially when the number  $n$  is greater than 2). This suggests that even a low number of sub-cluster members can also achieve good estimation accuracy.

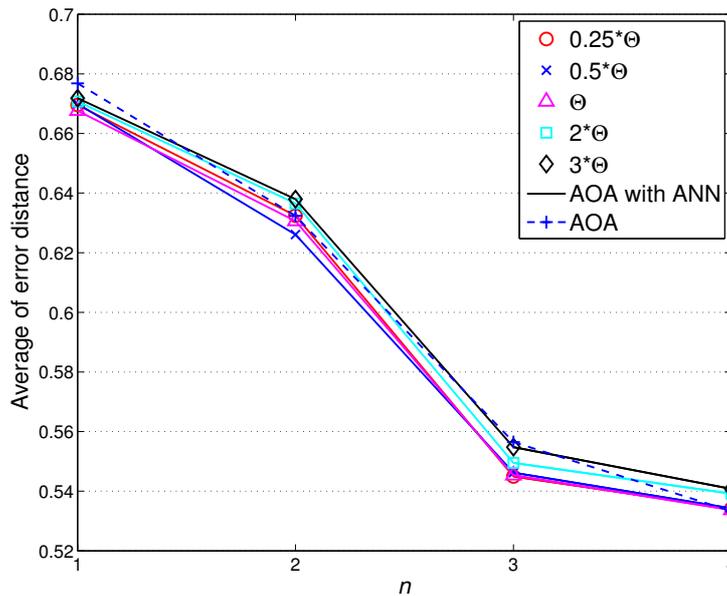
**Figure 12.** The target localization error (left); the estimation error with  $n = 2$  and various ratios of  $R/R_e$  (right).



As illustrated in Figure 12(left), the radio transmission range  $R$  is assumed to be the same as the target detection range  $R_e$ . Here the effect of varying target detection ranges on the performance is investigated with changing the ratio of the radio transmission range to target detection range. Figure 12(right) depicts that when the ratio is greater than one (*i.e.*,  $R/R_e > 1$ ), the sensors may fail to detect most target events and a larger network density may be required to detect the target of small signal magnitude and to suppress the estimation error. However, the estimation error decreases dramatically when the ratio  $R/R_e \leq 1$  due to sufficient detection coverage.

Notice that the above performance evaluation is based on AOA information only without executing estimation refinement. Figure 13 shows the comparison of target localization error with/without ANN refinement. With a moderate value of  $\theta_{th}$  (e.g.,  $0.25\Theta \leq \theta_{th} \leq \Theta$  with  $\Theta = 9.98$  degree), the proper refined sample area results in an improvement of estimation accuracy. However, loose NN-based angle information (e.g.,  $\Theta \leq \theta_{th} \leq 3\Theta$ ) may generate a broader sample area, which degrades the estimation performance.

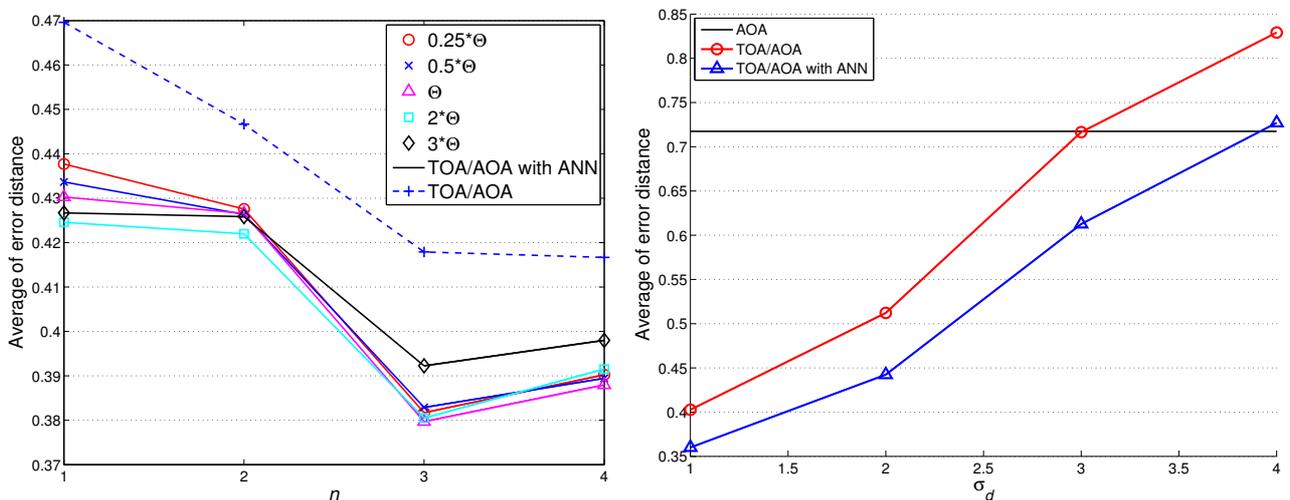
**Figure 13.** The comparison of target localization error with/without ANN refinement.



6.2.2. TOA/AOA with/without ANN Refinement

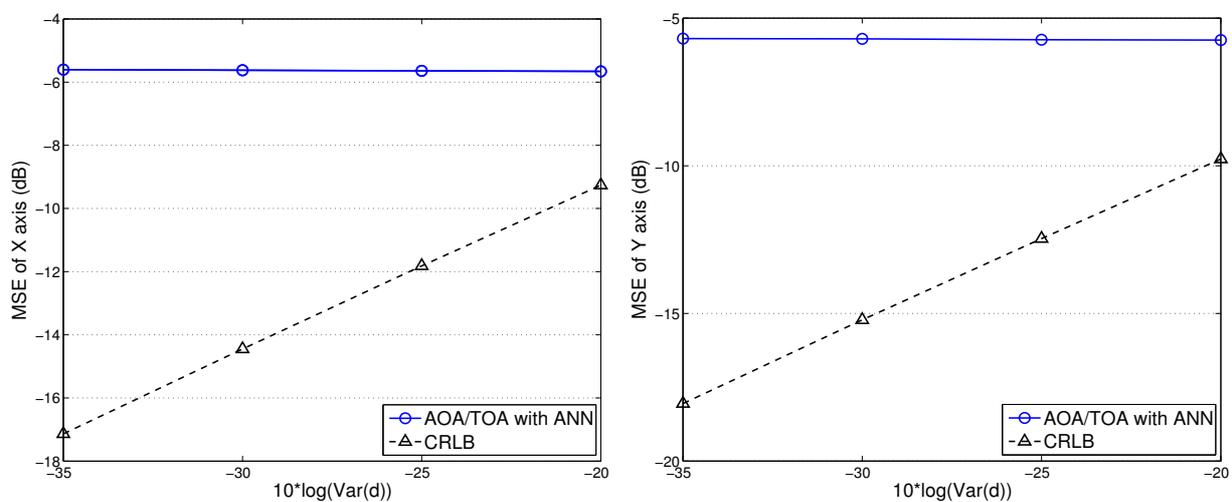
Compared with Figure 13, Figure 14(left) shows that the average localization error is significantly suppressed by applying the TOA information. Considering  $n = 2$  and  $\theta_{th} = 9.98$  degree, the average localization error of TCAT using AOA/TOA information with ANN refinement is about 36% less than that of TCAT using AOA information with ANN refinement. Nonetheless, as the deviation of TOA measurement increases, the localization performance of TCAT with joint TOA/AOA is approaching to that of TCAT with AOA only (Figure 14(right)). This is attributed to the fact that a noisy TOA measurement may lead to a broader sample area (*i.e.*, a lower particle density) for Bayesian filtering, which enlarges the localization error.

**Figure 14.** The comparison of target localization error with/without ANN refinement (**left**); the performance comparison of TCAT with joint AOA/TOA and TCAT with AOA only with varying the deviation of TOA measurement (**right**).



Given the variance of angle estimation  $\sigma_\theta = 0.5$  and with varying the uncertainty of distance estimation  $\sigma_d$ , here we consider the best achievable performance and the tracking performance with TCAT using TOA/AOA information. Notice that the CRLB and the performance of the proposed method tend to merge together with an increasing measurement uncertainty  $\sigma_d$ . Due to a small sample size for particle filtering and the limited capability of a sensor, as shown in Figure 15, fundamental problems when locating mobile target in a network are to estimate the distance between the reference sensors and the target and to determine the angle of arrival of the signals since accurate location estimates highly rely on precise TOA/AOA measurements and the processing capability of a sensor node.

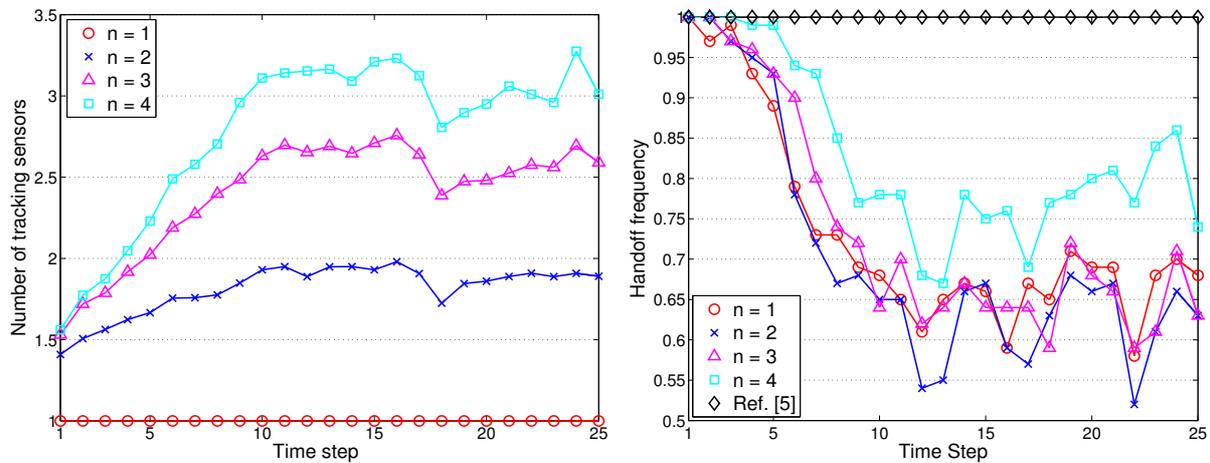
**Figure 15.** The performance comparison of TCAT with joint AOA/TOA and CRLB: X-axis (left) and Y-axis (right).



### 6.3. Protocol Characteristics

Figure 16(left) shows the typical runs of sub-cluster formation with  $N_S = 100$  and  $R/R_e = 1$ . Notice that the TCAT effectively organizes the sensors into tracking groups. Referring to Figures 12 (left) and Figure 16(right), observe that compared with the protocol in [5], the TCAT has a lower leader handoff frequency and there is no significant performance degradation during the leader handoff period. Moreover, considering different sub-cluster size  $n$  and a network with random uniform deployment, Figures 12(left) suggests that compared with the TCAT with  $n = 2$ , the TCAT with a smaller value of  $n$  (e.g.,  $n = 1$ ) with respect to handoff condition 1 and the TCAT with a larger value of  $n$  (e.g.,  $n = 3, 4$ ) with respect to handoff condition 2 lead to a higher frequency of inter-cluster handoff. Thus, the TCAT with  $n = 2$  may provide flexibility and robustness for distributed sensor scheduling management.

**Figure 16.** The typical runs of sub-cluster formation (**left**); the frequency of leader handoff (**right**).



6.4. Network Energy Consumption

The simulation is performed with different density of nodes, considering the number of messages transmitted and received involved in clustering and target localization. In order to evaluate the network performance, several models for measuring the energy dissipation per transmitted bit have been proposed [36,37]. Here, the energy model presented in [36] is applied to describe the energy dissipation. Assume the hardware energy dissipation is as follows [36]:

$$E_{Tx} = \begin{cases} tE_{elec} + t\varepsilon_{fs}d^2, & d < d_o \\ tE_{elec} + t\varepsilon_{mp}d^4, & d \geq d_o \end{cases} \tag{24}$$

$$E_{Rx} = tE_{elec} \tag{25}$$

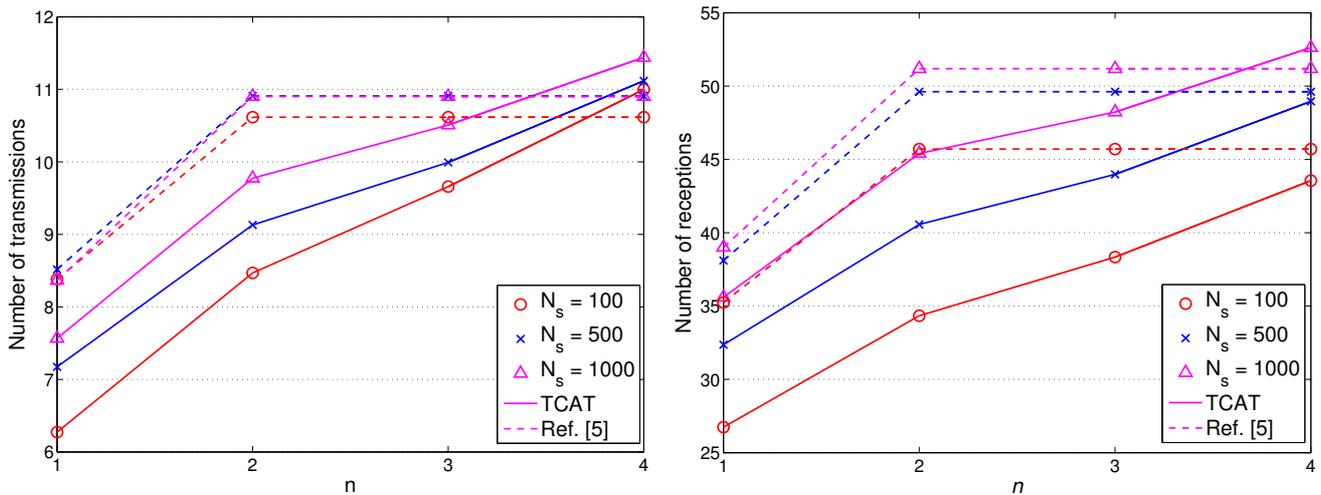
where  $E_{Tx}$  and  $E_{Rx}$  are energy consumption of a transmitter and a receiver, respectively,  $t$  is the data packet size,  $E_{elec}$  denotes the energy consumption of the electronic circuitry,  $\varepsilon_{fs}$  and  $\varepsilon_{mp}$  depend on distance  $d$  between the transmitter and the receiver for maintaining an acceptable bit-error rate, and  $d_o$  is a threshold of the transmission. The values of simulation parameters are detailed in Table 3 [36].

**Table 3.** The Values of Simulation Parameters [36].

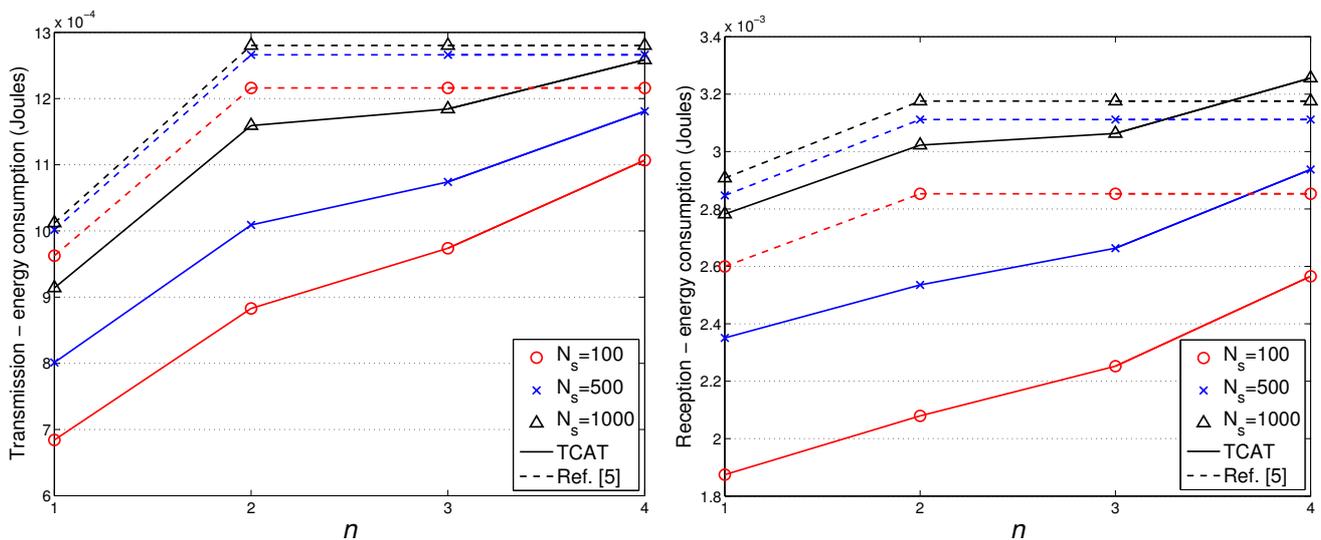
Parameter	Value
$E_{elec}$	50 nJ/bit
$\varepsilon_{fs}$	10 pJ/bit/m <sup>2</sup>
$\varepsilon_{mp}$	0.0013 pJ/bit/m <sup>4</sup>
$d_o$	Transmission range $R$
Energy for data aggregation	5 nJ/bit/signal
Data packet size $t$	2048 bits

Figures 17 and 18 show the accumulated energy consumption comparison between the TCAT scheme and the method in [5]. Observe that the number of transmissions/receptions grows nearly linearly as the tracking sub-cluster size increases. Referring to Figure 17, compared with a network with  $N_S = 100$  and  $1 \leq n \leq 3$ , a network with a larger scale (e.g.,  $N_S = 500, 1000$ ) may have a larger cluster size, which may lead to a higher number of transmission/reception for data gathering at each round and result in a faster network resource depletion. Nonetheless, the performance of TCAT is still superior to that of the approach in [5].

**Figure 17.** The number of transmission/reception: the communication cost.



**Figure 18.** The comparison of accumulated network energy consumption.



Given  $n = 2$ ,  $N_S = 100$ , and  $R/R_e = 1$ , Figure 12(right) shows that the tracking accuracy of TCAT is comparable to that of [5]. Moreover, as depicted in Figure 18(left), the energy consumption of transmissions with TCAT is about 26.4% less than that of [5] and in Figure 18(right) the energy consumption of receptions with TCAT is about 25% less than that of [5], which implies that the scheme in [5] may lead to a fast network energy depletion. Accordingly, the TCAT provides better network service characteristics compared to the protocol of [5].

Observe that in Figures 12 and 18, although the performance of TCAT with  $n = 1$  leads to a lower network energy consumption, compared with those of TCAT with  $n = 2 \sim 4$ , it results in a larger target localization error. Moreover, due to high correlation of sensing data in time and spatial domains, the sub-cluster with  $n > 2$  members may lead to undesired sensing redundancy. Therefore, considering the trade-off between performance and network energy consumption, the TCAT with  $n = 2$  may be a good choice for the tracking task.

## 7. Conclusions

Because of the resource-constrained sensors, feasible wireless sensor-based tracking systems require more breakthroughs in terms of network architecture, system design, and data processing techniques. In order to achieve good tracking quality, the number of sensors chosen for target positioning may be dynamically adjusted based on the available target and sensor information. Thus, incorporating the target motion information into cooperative positioning schemes with multiple sensors may be a good strategy to improve the estimation accuracy. Future plans will involve generalizing the method to implement a prototype of the tracking system, evaluate the merits of different cooperative schemes, explore the characteristics of target mobility model, and further examine the impact of target motion information on cooperative estimation performance.

## References

1. Bhatti, S.; Xu, J. Survey of Target Tracking Protocols using Wireless Sensor Network. In *Proceedings of the 2009 Fifth International Conference on Wireless and Mobile Communications*, Cannes La Bocca, France, 23–29 August 2009.
2. Gordon, N.; Salmond, D.; Smith, A.F.M. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. Radar Signal Process.* **1993**, *140*, 107–113.
3. Arulampalam, M.S.; Maskell, S.; Gordon, N.; Clapp, T. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Trans. Signal Process.* **2002**, *50*, 174–188.
4. Julier, S.; Uhlmann, J.K. General decentralized data fusion with Covariance Intersection(CI). In *Handbook of Multisensor Data Fusion*; Hall, D., Llinas, J., Eds.; CRC Press: Boca Raton, FL, USA, 2001.
5. Zou, Y.; Chakrabarty, K. Target localization based on energy considerations in distributed sensor networks. *Ad Hoc Netw.* **2003**, *1*, 261–272.
6. Chu, M.; Haussecker, H.; Zhao, F. Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. *Int. J. High Perform. Comput. Appl.* **2002**, *16*, 90–110.
7. Wang, H.; Pottie, G.; Yao, K.; Estrin, D. Entropy-based Sensor Selection Heuristic for Target Localization. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, New York, NY, USA, 26–27 April 2004; pp. 36–45.
8. Xiao, W.; Zhang, S.; Lin, J.; Tham, C.-K. Energy-efficient adaptive sensor scheduling for target tracking in wireless sensor networks. *J. Control Theory Appl.* **2010**, *2*, 86–92.

9. Zhang, S.; Xiao, W.; Ang, M.H.; Tham, C.-K. IMM Filter Based Sensor Scheduling for Maneuvering Target Tracking in Wireless Sensor Networks. In *Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information*, New York, NY, USA, 3–6 December 2007; pp. 287–292.
10. Liu, Y.; Xu, B.; Feng, L. Distributed IMM Filter Based Dynamic-Group Scheduling Scheme for Maneuvering Target Tracking in Wireless Sensor Network. In *Proceedings of the 2nd International Congress on Image and Signal Processing*, New York, NY, USA, 17–19 October 2009; pp. 1–6.
11. Suganya, S. A Cluster-based Approach for Collaborative Target Tracking in Wireless Sensor Networks. In *Proceedings of the First International Conference on Emerging Trends in Engineering and Technology-ICETET 2008*, Nagpur, India, 16–18 July 2008; pp. 276–281.
12. Zhao, F.; Shin, J.; Reich, J. Information-driven dynamic sensor collaboration for tracking applications. *IEEE Signal Process. Mag.* **2002**, *19*, 61–72.
13. Wei, J.; Liu, H.; Yu, Z. An energy-efficient target tracking framework in wireless sensor networks. *EURASIP J. Adv. Signal Process.* **2009**, *2009*, 1–14.
14. Wu, X.; Huang, G.; Tang, D.; Qian, X. Part II, CCIS 135. In *A Novel Adaptive Target Tracking Algorithm in Wireless Sensor Networks*; Chen, R., Ed.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 477–486.
15. Oka, A.; Lampe, L. Distributed target tracking using signal strength measurements by a wireless sensor network. *IEEE JSAC* **2010**, *28*, 1006–1015.
16. He, T.; Krishnamurthy, S.; Stankovic, J.A.; Abdelzaher, T.; Luo, L.; Stoleru, R.; Yan, T.; Gu, L.; Zhou, G.; Hui, J.; Krogh, B. VigilNet: An integrated sensor network system for energy-efficient surveillance. *ACM Trans. Sens. Netw.* **2006**, *2*, 1–38.
17. Xiao, W.; Xie, L.; Lin, J.; Li, J. Multi-Sensor Scheduling for Reliable Target Tracking in Wireless Sensor Networks. In *Proceedings of the 6th International Conference on ITS Telecommunications*, New York, NY, USA, 21–23 June 2006; pp. 996–1000.
18. Lin, J.; Xiao, W.; Lewis, F.; Xie, L. Energy-efficient distributed adaptive multisensor scheduling for target tracking in WSN. *IEEE Trans. Instrum. Meas.* **2009**, *58*, 1886–1896.
19. Yang, X.-J.; Xing, K.-Y.; Shi, K.-L.; Quan, P. Dynamic collaborative algorithm for maneuvering target tracking in sensor networks. *Acta Autom. Sinica* **2007**, *33*, 1030–1035.
20. Ma, H.; Ng, B.W.-H. Distributive Target Tracking in Wireless Sensor Networks under Measurement Origin Uncertainty. In *Proceedings of the Intelligent Sensors, Sensor Networks and Information*, Melbourne, Australia, 3–6 December 2007; pp. 299–304 .
21. Liu, Y.; Xu, B.; Feng, L. Energy-balanced multiple-sensor collaborative scheduling for maneuvering target tracking in wireless sensor networks. *J. Control Theory Appl.* **2011**, *9*, 58–65.
22. Williams, J.L.; Fisher, J.W., III; Willsky, A.S. Optimization Approaches to Dynamic Routing of Measurements and Models in a Sensor Network Object Tracking Problem. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '05*, Philadelphia, PA, USA, 18–23 March 2005; pp. 1061–1064.
23. Wen, C.Y.; Sethares, W.A. Automatic decentralized clustering for wireless sensor networks. *EURASIP J. Wirel. Commun. Netw.* **2005**, *5*, 686–697.

24. Savvides, A.; Han, C.C.; Srivastava, M.B. Dynamic Fine-grained Localization in *Ad-Hoc* Networks of Sensors. In *Proceedings of ACM SIGMOBILE*, Rome, Italy, 16–21 July 2001; pp. 166–179.
25. Zabiri, H.; Mazuki, N. A black-box approach in modeling valve stiction. *Int. J. Math. Phys. Eng. Sci.* **2010**, *4*, 9–16.
26. Qahwaji, R.; Colak, T. Neural Network-Based Prediction of Solar Activities. In *Proceedings of CITSA2006*, Orlando, FL, USA, 2006.
27. Kumar, S. *Neural Networks: A Classroom Approach*; McGraw-Hill: Singapore, 2005.
28. Zhao, F.; Guibas, L.J. *Wireless Sensor Networks: An Information Processing Approach*; Morgan Kaufmann: San Francisco, CA, USA, 2004.
29. Pinheiro, P.; Lima, P. Bayesian Sensor Fusion for Cooperative Object Localization and World Modeling. In *Proceedings of the 8th Conference on Intelligent Autonomous Systems*, Amsterdam, The Netherlands, 10–13 March 2004; pp. 1–8.
30. Aspnes, J.; Eren, T.; Goldenberg, D.K.; Morse, A.S.; Whiteley, W.; Yang, Y.R.; Anderson, B.D.O.; Belhumeur, P.N. A theory of network localization. *IEEE Trans. Mob. Comput.* **2006**, *5*, 1663–1678.
31. So, H.C.; Shiu, E.M.K. Performance of TOA-AOA hybrid mobile location. *IEICE Trans. Fundam.* **2003**, *E86-A*, 2136–2138.
32. Pottie, G.J.; Kaiser, W.J. Wireless integrated network sensors. *Commun. ACM* **2000**, *43*, 51–58.
33. Pister, K. Smart Dust (Keynote Address). In *Proceedings of the 2nd Int. Workshop Information Processing in Sensor Networks (IPSN)'03*, Palo Alto, CA, USA, 22–23 April 2003.
34. Chipcon CC2420 datasheet, Texas Instruments, 2007. Available online: <http://focus.ti.com/lit/ds/symlink/cc2420.pdf> (accessed on 7 November 2012).
35. Santi, P. *Topology Control in Wireless Ad Hoc and Sensor Networks*; John-Wiley & Sons: Chichester, UK, 2005.
36. Heinzelman, W.R.; Chandrakasan, A.; Balakrishnan, H. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In *Proceedings of IEEE HICSS*, Maui, HI, USA, 4–7 January 2000.
37. Bravos, G.; Kanatas, A.G. Energy Consumption and Trade-offs on Wireless Sensor Networks. In *Proceedings of the 16th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC'05*, Berlin, Germany, 11–14 September 2005; Volume 2, p. 1279.