An Energy-Efficient Node Selection Algorithm in Bearings-Only Target Tracking Sensor Networks

Xiaoxiao Jiang, Wei He, Xuguang Yang, and Yingguan Wang

Key Lab of Wireless Sensor Networks and Communication, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai 201800, China

Correspondence should be addressed to Wei He; hewei@mail.sim.ac.cn

Received 23 October 2013; Revised 30 December 2013; Accepted 6 January 2014; Published 16 February 2014

Academic Editor: Long Cheng

This paper discusses a node selection problem for bearings-only tracking in wireless sensor networks (WSNs). Saving energy and prolonging the lifetime of the network are the research focuses due to the severely constrained resource of WSNs. An energy-efficient network management strategy is necessary to achieve good tracking performance at low cost. In this paper, an energy-efficient node selection algorithm for bearings-only sensors in decentralized sensor networks is proposed. The residual energy of a node is incorporated into the objective function of node selection. A new criterion of node selection is also made to coordinate with the objective function. Compared with the other common methods, the proposed method can reduce the cost of the entire network, balance nodes energy expenditure, and extend the lifetime of network. Simulation results prove the effectiveness of the proposed method and show good performance in tracking accuracy and energy consumption.

1. Introduction

Target tracking has been widely applied in many fields of wireless sensor networks (WSNs), such as battle space surveillance, environment monitoring, and forwarning control. Because of the advantage of the concealment, passive technique becomes an important research direction of target tracking [1–4]. For passive target tracking in a wireless sensor network (WSN), the target state estimation of its position and velocity, which is the major task of target tracking, relies on the collaboration of sensor nodes. Due to extremely constrained resources especially the battery power of WSNs, it is necessary for an energy-efficient network management strategy to be implemented in the network. The main emphasis of network management is the study of the node selection problem (NSP). Therefore, this paper aims at designing an energy-efficient node selection approach to reduce the energy consumption of network, balance nodes energy expenditure, and extend the lifetime of network.

NSP has been attracting much attention of research and application in WSNs [5–25]. The simplest approach is the closest (CLT) node method mentioned in [5]. This method selects the closest nodes which have the shortest distance to the target to participate in tracking. It has simple calculations but low tracking precision as a result of the neglect of the angular diversity of nodes. In [6], the global node selection (GNS) method is proposed, which minimizes the expected filtered mean squared position errors using a metric like the geometrical dilution of precision (GDOP) [7, 8]. GNS uses all nodes locations every time interval for selecting the optimal $M$ nodes, where $M$ is a given value of active nodes number. In order to reduce more energy consumption, the autonomous node selection (ANS) method is proposed as a modification of GNS in [9]. ANS only needs to use local node information to achieve node selection. An uncertainty bounded model is introduced in [10], which considers an uncertainty area of the target as information utility. This method is good in precision but intensive in calculation. These methods both take the information utility brought by nodes as the objective function, but they do not consider the energy factor into the NSP. In [11], a node selection scheme is proposed within the framework of particle filter, which uses a node clustering method for collaborative tracking with considering the energy costs and the remaining energy. This scheme is efficient to achieve energy saving with a tolerable tracking error. However, the cluster head needs to
consume extra energy because of selecting cluster members and controlling the tracking, and the node selection also neglects the angular diversity of nodes. In [12], a user selection scheme is presented to minimize the overhead energy consumed by cooperative spectrum sensing in a cognitive radio sensor network. This method can conserve energy and achieve reasonably acceptable spectrum sensing accuracy, but it only focuses on the sensor node with a cognitive radio. Zhao et al. [13] propose an information-driven sensor querying (IDSQ) and data routing approach, which employs a mixture of both information gain and cost as the objective function for node selection. This method introduces Mahalanobis distance as a measure of information utility and defines an entropy based information-theoretic measure. However, Mahalanobis distance can be only applied to the range sensors, and the entropy is difficult to compute in practice. Besides, an entropy-based sensor selection heuristic approach is presented in [14]. Although it shows good accuracy for target localization, this approach has high computational complexity. More details and algorithms about node selection can be found in [15–25].

This paper works at a distributed network consisting of bearings-only sensor nodes, that is, microphone sensors arrays, which give direction of arrival (DOA) estimations for tracking [26]. In the distributed system, there is no processing center for nodes to send their observations [27]. All information is also processed locally on nodes. The distributed network is considered in this paper for the following reasons. On one hand, the distributed system is insensitive to the changes of the network topology by nodes addition, remove, or failure. On the other hand, the transmission range is much shorter than the centralized network, thereby saving energy and cost. In this paper, we propose an energy-efficient node selection algorithm for bearings-only target tracking. The goal of our algorithm is to provide greatest improvement to tracking accuracy at the lowest cost and balance the energy consumption between nodes. Therefore, we redefine the information utility brought by nodes, incorporate the residual energy of nodes into NSP, and make a new criterion for node selection.

The rest of this paper is organized as follows. Section 2 introduces the system model, the decentralized extended Kalman filter (DEKF), and the foundation for node selection. In Section 3, the proposed method and other common methods (ANS, GNS, and CLT) are described in detail. The simulation results are given in Section 4. Finally, Section 5 concludes the paper.

2. Problem Formulation

The problem discussed in this paper is target tracking with bearings-only sensors in a WSN. The WSN is composed of nodes deployed randomly in a surveillance region. Each node includes an array of microphone sensors to achieve the DOA estimation of an acoustic target. To reduce energy consumption, the network needs to select a subset of nodes to observe the target and let other nodes sleep. Then the active nodes share their observations at every time interval, which we call a snapshot. In this paper, we make the following assumptions of the WSN. First, the network is fully connected, and the location of each node in coordinates is known. Second, all nodes are time synchronized. Third, the probabilities of detection and false alarm are one and zero, respectively. Finally, this paper only aims at one single target in 2D field.

2.1. Measurement Model. In order to better comprehend the tracking system, Figure 1 shows the measurement model of this system. At a given snapshot \( k \), each active node can give a bearing with the DOA estimation for the target. Assume that the node measurement error obeys an additive white Gaussian noise model. Thus, for the active node \( j \), the measurement \( Z^j_k \) at the snapshot \( k \) is given by

\[
Z^j_k = \tilde{Z}^j_k + \omega^j_k,
\]  

where \( \tilde{Z}^j_k \) denotes the true DOA for the node \( j \) and \( \omega^j_k \) is a zero-mean white Gaussian noise with standard deviation \( \sigma_j \).

According to the geometric relationship between the node location and the target position, \( \tilde{Z}^j_k \) can be described by the following equations:

\[
h_j(X_k) = \arctan\left(\frac{p_{x,k} - p_{x}^j}{p_{y,k} - p_{y}^j}\right)
\]

where \( X_k \) is the target state containing the position and the velocity of the target, \( h_j(X_k) \) represents the nonlinear function related to \( X_k \), which is the arc tangent function denoted by \( \arctan \), and \([p_x^j, p_y^j]\) is the location of the node \( j \) in Cartesian coordinates.

![Figure 1: The measurement model of the tracking system.](image-url)
2.2. Decentralized Extended Kalman Filter. After the active
nodes have estimated the DOA of the target, the tracking
system fuses all measurements of the active nodes to utilize
the extended Kalman filter (EKF) for tracking. The EKF
is viewed as the nonlinear counterpart of the classical
Kalman filter [28]. Because the decentralized system is
considered, this paper uses the decentralized extended Kalman
filter (DEKF) to complete the trace process.

The target state is denoted by \( X_k = [p_{x,k}, p_{y,k}, v_{x,k}, v_{y,k}]^T \),
where \([p_{x,k}, p_{y,k}]^T\) represent the x-axis and y-axis position
components and \([v_{x,k}, v_{y,k}]^T\) represent the x-axis and y-
axis velocity components at the snapshot \( k \). Then the state
transition can be expressed as

\[
X_k = f(X_{k-1}, k) + B_k W_k,
\]

where \( f \) is a state transition function which is subject to the
target motion model, \( B_k \) represents the process noise
distribution, and \( W_k \) is the process noise matrix which
follows a zero-mean Gaussian distribution with covariance
matrix \( Q_k \).

Assume that there are \( M \) nodes participating in tracking
at the snapshot \( k \); then the measurements of all \( M \) nodes are
denoted by \( Z_k = [Z^1_k, Z^2_k, \ldots, Z^M_k]^T \), and hence \( Z_k \) can be
written as

\[
Z_k = h(X_k) + V_k
\]

\[
h(X_k) = [h_1(X_k), h_2(X_k), \ldots, h_M(X_k)]^T,
\]

where \( V_k = [\alpha_1, \alpha_2^2, \ldots, \alpha_M^2]^T \) is the measurement noise
matrix.

Like the Kalman filter, the DEKF also has two steps: the
prediction step and the update step. The DEKF is carried out
by the following equations.

(a) The prediction step:

\[
X_{k+1|k} = F_k X_k
\]

\[
P_{k+1|k} = F_k P_{k+1|k} F_k^T + B_k Q_k B_k^T,
\]

where \( F \) and \( P \), respectively, are the state transition matrix
and the state covariance matrix about \( X \). This step can be
implemented for the prediction of \( X \) and \( P \).

(b) The update step:

\[
V_j = \frac{1}{r_j} \left( -\sin \phi_j, \cos \phi_j, 0, 0 \right)^T
\]

\[
Y^j_{k+1} = \frac{1}{\sigma_j} V_j^T V_j
\]

\[
y^j_{k+1} = \frac{1}{\sigma_j^2} V_j^T \left[ Z^j_{k+1} - h_j(X_{k+1|k}) + V^T_j X_{k+1|k} \right]
\]

\[
P_{k+1|k+1}^{-1} = P_{k+1|k}^{-1} + \sum_{j=1}^M Y^j_{k+1}
\]

\[
X_{k+1|k+1} = P_{k+1|k+1} \left( P_{k+1|k+1}^{-1} X_{k+1|k} + \sum_{j=1}^M Y^j_{k+1} \right),
\]

where \( V_j \) is the gradient of \( h_j(\cdot) \) about \( X_{k+1|k} \) and \([r_j, \phi_j] \) is the
polar representation of the node \( j \) location \([p^j_x, p^j_y] \) relative to
the predicted target position \([p_{k+1|k}, \phi_{k+1|k}] \). In the update
step, all active nodes share their measurements and obtain the
updated target state and state covariance matrix by (6)–(10).

2.3. Foundation for Node Selection. The goal of NSP is to find
out the best set of active nodes from all nodes in the WSN
to track the target. In order to ensure the tracking accuracy,
the information utility is usually considered to be a measure
of the expected mean square position error. That is, if NSP
select the active nodes set which has the largest information
utility, then target tracking will get the least expected mean
square position error.

For example, the information utility of the active nodes
set \( S_a \) can be defined as the reciprocal of the posterior position
error \( \rho(S_a) \), and thus it can be written as

\[
U(S_a) = \frac{1}{\rho(S_a)}.
\]

According to the updated state covariance matrix in the
DEKF, \( \rho(S_a) \) can be given by

\[
\rho(S_a) = P_{k+1|k+1} \left( 1, 1 \right) + P_{k+1|k+1} \left( 2, 2 \right),
\]

where \( A(a,b) \) is the \( a \)-th row and the \( b \)-th column element of
the matrix \( A \), and then \( P_{k+1|k+1} \left( 1, 1 \right) \) and \( P_{k+1|k+1} \left( 2, 2 \right) \)
represent the position components of \( P_{k+1|k+1} \) which can be
derived from (9). For simplicity, the inverse matrix of
\( P_{k+1|k+1} \), \( P_{k+1|k+1}^{-1} \), is denoted by \( J_f \). Then (9) can be
rewritten as

\[
J_f = I_p + \begin{bmatrix} 0 & \rho(S_a) \\ 0 & 0 \end{bmatrix}
\]

\[
J_p = P_{k+1|k+1}^{-1}
\]

\[
J = \sum_{j \in S_a} \frac{1}{\sigma_j^2} \begin{bmatrix} \sin^2 \phi_j & -\sin \phi_j \cos \phi_j \\ -\sin \phi_j \cos \phi_j & \cos^2 \phi_j \end{bmatrix}
\]

Therefore, \( \rho(S_a) \) can be obtained as follows

\[
\rho(S_a) = \left( \text{tr}[J_p] + \text{tr}[J] \right)
\]

\[
\times \left( \det[J_p] + \det[J] \right)
\]

\[
+ \sum_{j \in S_a} \left[ \begin{bmatrix} \cos^2 \phi_j \\ \sin \phi_j \cos \phi_j \end{bmatrix} \begin{bmatrix} \cos^2 \phi_j \\ \sin \phi_j \cos \phi_j \end{bmatrix} \right]^{-1}
\]

\[
J_f = [J_p]_{1:2,1:2} - [J_p]_{1:2,3:4} [I_p]_{3:4,3:4}^{-1} [J_p]_{3:4,1:2},
\]

where \([I_p]_{a:b,c:d} \) is the submatrix of \( J_p \) which consists of rows
\( a \) through \( b \) and columns \( c \) through \( d \). The detailed derivation
process can be referred to [6].
3. Node Selection for Tracking

The aim of node selection is to select an active nodes subset $S_a$ from the candidate nodes set $S$ that achieves better tracking accuracy with less cost of network at every snapshot. In the WSN, the nodes possess limited energy and lifetime, so balancing energy consumption strategy must be considered to prolong the lifecycle of network. This paper presents an energy-efficient node selection (EENS) algorithm, which is required to find the best set $S_a$ with up to $M$ active nodes, where $M$ is the constant number at each time snapshot. This section first introduces the system initialization process. Then, it describes the detailed steps of the EENS algorithm. Finally, it gives the summary of the ANS, GNS, and CLT algorithms.

3.1. System Initialization. Once a target appears in the monitoring area, the nodes deployed in this area will detect the target and give a bearing estimation of the target position. Then the tracking system should decide which nodes will participate in tracking. The objective of choosing nodes is minimizing the mean square localization error. Therefore, we use the GDOP metric to be a method of system initialization.

Assume that there are $N_a$ nodes in the WSN. We need to find the best nodes set $S_a$ of cardinality $M$ from all $N_a$ nodes for tracking at the beginning. All nodes have already known the target position by the localization algorithm, which is not the content of this paper. Then, the posterior mean square position error by GDOP can be given by

$$\rho (S_a) = \frac{\text{tr} [J]}{\det [J]}.$$  \hspace{1cm} (16)

It can be seen that this equation is a simple expression of (14) where the prior information is ignored so that $\dot{J}_p = 0$.

Therefore, the problem is to find the minimum of $\rho(S_a)$, thereby determining which nodes compose the active set $S_a$ to observe the target at the next snapshot. Consider that all nodes communicate their locations with each other. Here, there are two methods to search the best nodes set for reference. One method is adopting an exhaustive strategy to try all combinations. However, this method has high computational complexity. For example, if the system decides to select $M$ active nodes, then the number of computations is

$$n_c = \frac{N_a!}{M! \times (N_a - M)!}.'$$ \hspace{1cm} (17)

The other method is "add one node at a time" (AOAT), where one node is chosen to be activated at a time. The details can be seen from [6]. It is minimizing the mean squared position error when adding a new node to $S_a$. The system initialization process is shown in Figure 2.

3.2. EENS. After system initialization, the active nodes in $S_a$ have already been selected to track the target at the current snapshot $k$. At the same time, other nodes not belonging to $S_a$ go to sleep and keep a dormant mode. The system fuses the nodes measurements of $S_a$ so that $X_{d,k}$ and $P_{d,k}$ can be obtained by the updated steps of DEKF. Then the prediction of the target position is calculated for the following node selection. The EENS can be described from two perspectives, the active nodes and the idle nodes. More specifically, the first stage is that EENS selects a nodes subset $S_a^c$ with desired cardinality $N_d$ from $S_a$ to keep the active status, and the second stage is that it selects $\kappa$ idle nodes to be activated at the snapshot $k + 1$.

3.2.1. The First Stage. The main task of this stage is to find the best active subset of $N_d$ nodes from $S_a$ that maximizes (11). Generally the nodes which are active at the current snapshot are also best for tracking at the next snapshot, because the target moving distance is so small that the network topology only changes a little. Therefore, the performance of node selection algorithms is affected by the parameter setting of $N_d$. A detailed explanation of this problem is beyond the scope of this paper. This stage still utilizes the AOAT method to choose the best active nodes set labeled as $S_a^c$ while the unselected nodes are denoted as $\bar{S}_a$.

3.2.2. The Second Stage. The goal of EENS is to find out the active nodes set which can bring the largest information utility, save the cost of the network, and balance energy expenditure on each node. After the first stage, it is possible that an idle node can be better than a node in $S_a^c$ to join $S_a^c$. Therefore, the problem is to find the minimum of $\rho(S_a^c)$, thereby determining which nodes compose the active set $S_a^c$ to observe the target at the next snapshot. Consider that all nodes communicate their locations with each other. Here, there are two methods to search the best nodes set for reference. One method is adopting an exhaustive strategy to try all combinations. However, this method has high computational complexity. For example, if the system decides to select $M$ active nodes, then the number of computations is

$$n_c = \frac{N_a!}{M! \times (N_a - M)!}.'$$ \hspace{1cm} (17)

The other method is "add one node at a time" (AOAT), where one node is chosen to be activated at a time. The details can be seen from [6]. It is minimizing the mean squared position error when adding a new node to $S_a$. The system initialization process is shown in Figure 2.
Therefore, EENS needs to calculate the information utility of idle nodes $\nu_i(S_a)$, the criterion of node selection $\varphi_i(S_a)$ and select the $k$ largest ones to be activated. EENS implements two steps to select the idle nodes to become active. The details are as follows.

First, the active node $j$ in $S_a$ computes its information utility by

$$\nu_j(S_a) = U(S_a) - U(S_a \setminus \{j\}).$$

The maximum of the information utility of these nodes is denoted as the threshold $\Gamma$ for other nodes to join $S_a$. Then the active nodes in $S_a$ broadcast the target state prediction, their locations, and the threshold $\Gamma$.

Second, the idle nodes which receive information from all active nodes calculate their information utility $\nu_i(S_a)$ by

$$\nu_i(S_a) = U(S_a \cup \{i\}) - U(S_a).$$

The information utility $\nu_i$ is considered as an important metric for the idle node $i$ to add to $S_a$ to become active. The greater the information utility $\nu_i$ is, the more possible the node $i$ is to be activated.

However, the distance of the target motion is usually so small that the current active nodes are probably also best at the next snapshot. Therefore, some nodes are always selected in $S_a$, which makes these nodes consume energy faster. In order to balance energy expenditure on every node and extend the lifetime of the whole network, the residual energy of the node is introduced into the optimization of NSP. The criterion of NSP can be described as a mixture of information utility and residual energy. Then the criterion for the node $i$ to join $S_a$ is given by

$$\varphi_i(S_a) = \alpha \cdot \mu(\nu_i(S_a)) + (1 - \alpha) \cdot \mu(r_e_i),$$

where $r_e_i$ represents the residual energy of the node $i$, $\alpha$ is the relative weighting of the information utility and residual energy, and $\mu(\cdot)$ is the normalization function. Normalization is designed to let the information utility and residual energy of each candidate node have the same level of effects on $\varphi_i$. Therefore, the function $\mu(\cdot)$ is defined by

$$\mu(\nu_i(S_a)) = \frac{\nu_i(S_a)}{\sum_{i \in S_a} \nu_i(S_a)} \quad \text{and} \quad \mu(r_e_i) = \frac{r_e_i}{\sum_{i \in S_a} r_e_i}.$$  

The first term in $\varphi_i(S_a)$ indicates the usefulness of the measurements of the node $i$. The usefulness means the improvement of the position estimation accuracy by the node $i$. The second term characterizes the effort of the residual energy on the node selection. The criterion fully considered the two factors with the value of $\alpha$, and there is a bias toward the factor of a larger coefficient. In addition, the validity of this criterion can be proved by the two points: first, any one candidate node adding to $S_a$ can improve the information utility of the active nodes set, and the different nodes lead to different degree of the improvement, so the tracking performance can be ensured by the nodes in $S_a$; second, the introduction of the energy factor just make the node selection have a deflection but not affect the tracking correctness.

According to the quantity requirements for the nodes participants, we choose the nodes whose selected criterion $\varphi_i(S_a)$ are the $k$ largest ones, and thus the objective function can be defined as

$$i = \arg \max_{i \in S_a} \{\varphi_i(S_a), k\}.$$  

In addition, the number of active nodes $N_a$ is a function of the parameters $N_d$ and $\kappa$ at every snapshot. Because of the assumption that the active nodes number is up to $M$ as mentioned above, the parameter $\kappa$ can be obtained from

$$\kappa = \begin{cases} M - N_d, & N_c > M - N_d \\ N_c, & N_c < M - N_d \end{cases},$$

where $N_c$ is the nodes number of the candidate set $S_c$. In other words, $\kappa = \min[N_c, M - N_d]$. Then, the active nodes number can be computed by

$$N_a = N_d + \kappa.$$  

Now the set $S_a$ is relabeled as $S_a^*$, which is the active nodes set at the next snapshot. Figure 3 shows the process of the EENS in a snapshot interval for node selection.

Furthermore, the utility $\nu_i$ is bounded above by a function of the distance between the target and the idle node $i$, which can be computed by

$$\nu_i(S_a) < \frac{1}{\sigma^2 r_i^2}.$$  

The proof can be seen in the appendix.

The upper bound of the information utility brought by an idle node can help to eliminate the nodes which are unable to be better than the nodes in $S_a^*$ and determine the candidate nodes set $S_c$. According to this upper bound, the distance between the idle node $i$ and the target should be

$$r_i < r_f = \frac{1}{\sigma_{\min} \sqrt{\Gamma}}.$$  

The nodes which can satisfy (26) are labeled as a candidate set $S_c$.

The broadcast range of an active node needs to be set high enough to cover other active nodes in $S_a^*$ and the candidate idle nodes to join $S_a^*$. Therefore, the broadcast range for the node $j \in S_a^*$ is

$$d_j = r_j + r_c$$

$$r_c = \max \left\{ r_c, \max_{j \in S_a} r_j \right\},$$

where $r_c$ is the bias to the factor of a larger coefficient.
where $r_c$ represents the critical range to the target to cover all active nodes and candidate nodes. In fact, only the nodes which fall in the communication range of the active nodes from the previous snapshot can be chosen to add to the next active nodes set. These available nodes form the candidate set $S_c$.

3.3. ANS and GNS. The ANS algorithm uses different criteria to select idle nodes to join $S_{−a}$. As our presented algorithm, ANS also first choose $N_d$ nodes from $S_a$ to keep active mode by minimizing $\rho(S_{−a})$, but it defines the differential utility of an active node $j$ in $S_a$ as

$$du(j) = U(S_{−a} \cup \{j\}) - U(S_{−a} \{j\})$$  \hspace{1cm} (28)

and the differential utility of an inactive node $i$ as

$$du(i) = \max_{j \in S_{−a}} [U(S_{−a} \cup \{j\}) - U(S_{−a} \{j\})].$$  \hspace{1cm} (29)

Then these nodes in $S_{−a}$ set a threshold $\tau$ for an inactive node to join the active set as the $y$th ($1 \leq y \leq N_d$) largest differential utility $du(j)$, $j \in S_a$. Active nodes broadcast the threshold, predicted target state and their location. Once the inactive nodes receive the data from the active set, they calculate their differential utility by (29). If the differential utility exceeds the threshold, then the node joins the active set. The ANS algorithm does not consider the energy factor in node selection.

Unlike ANS, although GNS incorporates the search stage of ANS, it uses all nodes within range of the $S_a$ active nodes to implement searching. In other words, GNS does not need to implement the steps of selecting the candidate nodes set and $N_a = N_d$.

3.4. CLT. All nodes in the WSN know each node location after system initialization, so the active nodes can select idle nodes to become active at the next snapshot by the information of their close nodes. The CLT method simply selects the $N_a$ nodes with the smallest virtual range to the target. The virtual range is defined as

$$Vr(i) = \sigma r_c.$$  \hspace{1cm} (30)

Then this method sorts the virtual range so that it can choose the nodes. This method is computationally simple. However, its tracking accuracy is possible to be reduced due to the ignorance of the angular diversity of nodes.

4. Simulation

In order to demonstrate the effectiveness of the proposed method for node selection, we apply it to the WSN for bearings-only target tracking and compare it with CLT, GNS, and ANS in terms of the complexity, the execution time, the root mean square error (RMSE) of the target position estimation, energy consumption, and residual energy statistics on individual nodes.

Consider a scene that 50 nodes are uniformly deployed over a monitoring area of $1000 \times 1000$ m$^2$. The target occurs at $[-400, -400]$ and moves through the monitoring area with a constant velocity (CV) model. Figure 4 has shown the target trajectory with initial velocity of 20 m/s and nodes deployment in one trial. To illustrate the performance of different methods, 1000 Monte Carlo trials are generated for CLT, GNS, ANS, and EENS, respectively. The measurement noise of each node is a zero-mean Gaussian white noise with standard deviation $5^\circ$. For simplicity, we only consider the situation without prior information ($\tilde{J}_p = 0$). Here, the relative weighting parameter $\alpha$ of EENS is considered as 1 to testify the effectiveness of our proposed method. The effect of $\alpha$ on the node selection will be evaluated later.

Table I gives the computational complexity and the execution time of CLT, GNS, ANS, and EENS. The initial node selection at the beginning of target tracking uses the same method, as the system initialization in Section 3.1. Therefore, the comparison of the computational complexity focuses on the node selection at one time step. In other words, the
The target trajectory and nodes deployment.

![Figure 4: The target trajectory and nodes deployment.](image)

**Table 1**: The computational complexity and the execution time of CLT, GNS, ANS, and EENS.

| Methods | Total computations | Execution time/s |
|---------|--------------------|------------------|
| CLT     | $N_a \times N_s$   | 0.031            |
| GNS     | $\frac{N_a \times (N_s - 1)}{2} + \sum_{m=2}^{N_s-1} (N_a - m)$ | 0.398            |
| ANS     | $\frac{N_a \times (N_a - 1)}{2} + \sum_{m=2}^{N_a-1} (N_a - m) + N_c$ | 0.058            |
| EENS    | $\frac{N_a \times (N_a - 1)}{2} + \sum_{m=2}^{N_a-1} (N_a - m) + N_c$ | 0.048            |

The total computations of node selection are calculated from the current snapshot to the next snapshot. The execution time is the average tracking time of 1000 Monte Carlo trials. The parameter setting is $N_d = 2$ and $M = 5$. As can be seen from Table 1, GNS has the highest complexity, CLT has the simplest computations, and ANS and EENS have a satisfactory performance on the computational complexity and execution time.

Figure 5 has shown the tracking performance comparison of four different node selection methods versus the average number of active nodes per snapshot. The average RMSE for Monte Carlo simulations can be seen from different symbols. The RMSE of CLT is far larger than the other three methods and EENS performs better than other methods overall. In addition, EENS can use less active nodes but achieve almost the same tracking accuracy or better than the other methods.

The energy consumption results are compared in Figures 6–9. In this simulation, we refer to the communication model for WSNs in [29]. The energy to implement operations is far smaller than the energy to transmit data, so the simulation ignores the energy for calculations. Assume that the energy to transmit $l$ bits of data a distance of $d$ meters is

$$E_t = l \varepsilon_{\text{elec}} + l \varepsilon_{\text{amp}} d^4$$

and the energy to receive the data is

$$E_r = l \varepsilon_{\text{elec}},$$

where $\varepsilon_{\text{elec}}$ represents the energy per bit to run the electronics, and $\varepsilon_{\text{amp}}$ is the energy per bit to run the power amplifier. Given $\varepsilon_{\text{elec}} = 0.5 \times 10^{-7}$ J/bit, $\varepsilon_{\text{amp}} = 1.3 \times 10^{-14}$ J/bit/m$^4$ and $l = 500$ bits. Figure 6 shows the average energy consumption of the entire network per tracking process against the average number of active nodes. The comparison result of CLT, GNS, ANS, and EENS is shown in Figure 6(a). Consistent with the practical experience, the network costs more energy as the active nodes number increases. It can be seen that CLT consumes the maximum amount of energy and GNS is the second. This is because when the network uses CLT and GNS for node selection, the nodes need to communicate with each other to get the knowledge of all nodes with more power consumption. However, the nodes in ANS and EENS only use their local information, so ANS and EENS can save much energy and do not be affected once the network topology changes. In order to see the difference between ANS and EENS more clearly, Figure 6(b) shows the enlarged figure of the rectangular portion with the green dashed line in Figure 6(a). From Figure 6(b) we can see that EENS uses less energy consumption than ANS.

Figure 7 gives out the average energy consumption of the entire network per tracking process against RMSE. Similarly to Figure 6, Figure 7(a) shows the comparison result of the four different algorithms and Figure 7(b) shows the enlarged figure of the rectangular portion with the green dashed line in Figure 7(a). From Figure 7(a) it can be seen that ANS and
EENS can expend substantially less energy with meeting the requirements of tracking accuracy. To better see the performance comparison of ANS and EENS, Figure 7(b) shows that EENS has a significant decrease of energy consumption.

Note here that there is each inflection point in Figures 6(b) and 7(b). Its presence does not comply with the common apperception rule. The reason is the refreshing phenomenon which occurs in ANS when an active set of two nodes becomes collinear with the target [9], which leads to abundant nodes to join target tracking. For example, when the active nodes number is about 3 (the parameter $N_d$ of ANS is general to be 2), ANS easily makes a large amount of the nodes to be activated and consumes more energy, so the left-most point of ANS in Figure 6(b) is higher than the next point in the $y$ axis. In the same way, the right-most point of ANS in Figure 7(b) can be explained.
Figures 8 and 9 have shown the average energy consumption per node against the average number of active nodes and RMSE, respectively. The results of these two figures are in accord with Figures 6 and 7.

The above simulation results obviously show that the EENS method is effective and energy efficient. However, these simulations set the parameter $\alpha = 1$ in EENS. In order to verify the role of $\alpha$ in EENS, we simulate EENS with $\alpha = 0.2$, $\alpha = 0.5$, $\alpha = 0.8$, and $\alpha = 1$ to compare the performance for 1000 Monte Carlo trials. Because of the relatively good performance of ANS compared to CLT and GNS, we use ANS as the reference to the different parameter settings of EENS. Table 2 gives out the performance comparison results with $N_d = 2$ and $M = 5$. It shows that EENS with different $\alpha$ also have good tracking accuracy and energy efficiency. However, the performance of EENS with different $\alpha$ does not appear regular change as $\alpha$ gets larger. The impact of $\alpha$ makes the node selection bias towards the larger weight.

Figure 8: The average energy consumption per node against the average number of active nodes. (a) The comparison result of CLT, GNS, ANS, and EENS. (b) The enlarged figure of the rectangular portion with green dashed line in (a).

Figure 9: The average energy consumption per node against RMSE. (a) The comparison result of CLT, GNS, ANS, and EENS. (b) The enlarged figure of the rectangular portion with green dashed line in (a).
Table 2: The performance comparison of ANS and EENS with different parameter settings.

| Methods       | Active nodes number | RMSE (m) | Average energy consumption of network per tracking process (J) | Average energy consumption per node (J) |
|---------------|---------------------|----------|---------------------------------------------------------------|--------------------------------------|
| ANS           | 3.0412              | 3.2084   | 45.632                                                        | 0.3001                                |
| EENS (\(\alpha = 0.2\)) | 3.3989              | 2.8497   | 5.7986                                                        | 0.0341                                |
| EENS (\(\alpha = 0.5\)) | 3.3961              | 2.8626   | 5.8021                                                        | 0.0342                                |
| EENS (\(\alpha = 0.8\)) | 3.3963              | 2.8510   | 5.8094                                                        | 0.0342                                |
| EENS (\(\alpha = 1\))  | 3.3961              | 2.8560   | 5.7975                                                        | 0.0341                                |

Figure 10: The residual energy statistics on individual nodes.

5. Conclusion

This paper proposed an energy-efficient node selection for bearings-only target tracking in WSNs. This method redefines the information utility of the idle node to join the active nodes set and introduces the factor of the residual energy into the objective function for node selection. It makes the criterion of choosing at most \(M\) nodes to participate in tracking. In the decentralized WSN, the proposed method only uses the local information of the node so it can save more energy and better accommodate to the change of networks. The incorporation of the factor of the residual energy makes the control of the entire network easier. According to the setting of the weighting parameter, the network can balance the energy consumption on nodes and extend the life of the network. Simulation results validated the good performance of the proposed method in terms of tracking RMSE, energy expenditure, and the residual energy statistics on individual nodes.

Appendix

Proof of the upper bound of the information utility \(n_u(S_n)\).

From (19), the information utility \(n_u(S_n)\) is computed by the utility of the two sets; one is the active nodes set \(S_a\), and the other is

\[ S'_n = S_a \cup \{i\} \]  \hspace{1cm} (A.1)

According to (11)–(15), the information utility of \(S'_n\) can be written by the following equations:

\[ U(S'_n) = \frac{\text{det}(\tilde{J}_{f})}{\text{tr}(\tilde{J}_{f})} \]  \hspace{1cm} (A.2)

\[ \tilde{J}_{f} = \tilde{J}_{p} + J. \]

Because the matrix \(\tilde{J}_{f}\) is positive definite, its eigenvalues can be represented as

\[ \text{eigenvalue (1)} = \frac{\text{tr}(\tilde{J}_{f})}{2} (1 - \gamma), \]  \hspace{1cm} (A.3)

\[ \text{eigenvalue (2)} = \frac{\text{tr}(\tilde{J}_{f})}{2} (1 + \gamma), \]

where \(\gamma\) is the angular diversity parameter for \(S_n\). Therefore,

\[ \text{det}(\tilde{J}_{f}) = \frac{\text{tr}(\tilde{J}_{f})^2}{4} (1 - \gamma^2). \]  \hspace{1cm} (A.4)

Hence, the information utility of \(S'_n\) can be rewritten as

\[ U(S'_n) = \frac{\text{tr}(\tilde{J}_{f})}{4} (1 - \gamma^2). \]  \hspace{1cm} (A.5)
Similarly, the information utility of $S_n^a$ is

$$U(S_n^a) = \frac{\det \{\tilde{J}_f\}}{\text{tr} \{\tilde{J}_f\}}, \quad (A.6)$$

$$\tilde{J}_f = \tilde{J}_f + \frac{1}{\alpha_i^2 r_i^2} n_i n_i^T = UU^T + \frac{1}{\alpha_i^2 r_i^2} n_i n_i^T,$$

where $U$ is an orthonormal matrix, $\Lambda$ is a two-diagonal matrix by the above two eigenvalues, and $n_i = [-\sin \phi_i, \cos \phi_i]^T$. Then,

$$\tilde{n}_i^T U = [b, \pm \sqrt{1-b^2}]. \quad (A.7)$$

where $-1 \leq b \leq 1$. Then,

$$U^T \tilde{J}_f U = \Lambda + \frac{1}{\alpha_i^2 r_i^2} [b^2 \left\{ \begin{array}{l} b \sqrt{1-b^2} \quad b \sqrt{1-b^2} \\ b \sqrt{1-b^2} \quad 1-b^2 \end{array} \right\}]. \quad (A.8)$$

Thus, the determinant and the trace of $\tilde{J}_f$ are

$$\det \{\tilde{J}_f\} = \frac{\text{tr} \{\tilde{J}_f\}^2}{4} (1 - \gamma^2) + \frac{\text{tr} \{\tilde{J}_f\}}{2} \frac{1}{\alpha_i^2 r_i^2} (1 + \gamma - 2b^2),$$

$$\text{tr} \{\tilde{J}_f\} = \text{tr} \{\tilde{J}_f\} + \frac{1}{\alpha_i^2 r_i^2}. \quad (A.9)$$

Then, the information utility of $nu_i(S_n^a)$ is

$$nu_i(S_n^a) = \left( \frac{\text{tr} \{\tilde{J}_f\}^2}{4} (1 - \gamma^2) + \frac{\text{tr} \{\tilde{J}_f\}}{2} \frac{1}{\alpha_i^2 r_i^2} \right) \left( \frac{\text{tr} \{\tilde{J}_f\} + \frac{1}{\alpha_i^2 r_i^2}}{1 + \gamma - 2b^2} \right)^{-1}$$

$$\times \left(1 + \gamma - 2b^2 \right) \left( \frac{\text{tr} \{\tilde{J}_f\} + \frac{1}{\alpha_i^2 r_i^2}}{1 + \gamma - 2b^2} \right) - \frac{\text{tr} \{\tilde{J}_f\}}{4} (1 - \gamma^2)$$

$$= \frac{\text{tr} \{\tilde{J}_f\} \cdot (1 + \gamma^2 - 4b^2)}{4 \cdot (\text{tr} \{\tilde{J}_f\} + \frac{1}{\alpha_i^2 r_i^2})} \cdot \frac{1}{\alpha_i^2 r_i^2}. \quad (A.10)$$

Because $0 \leq \gamma \leq 1, -1 \leq b \leq 1$, and $\text{tr} \{\tilde{J}_f\} > 0$, the value range of $nu_i(S_n^a)$ is

$$nu_i(S_n^a) < \frac{1}{\alpha_i^2 r_i^2}. \quad (A.11)$$

Then the upper bound is obtained.

**Conflict of Interests**

The authors declare that there is no conflict of interests regarding the publication of this paper.

**Acknowledgments**

This work was supported by the National Basic Research Program of China (973 Program) under Grant 2011CB302901 and the National Science and Technology Major Project under Grant 2014ZX03005001-002.

**References**

[1] Z. Wang, J. Luo, and X. Zhang, "A novel location-penalized maximum likelihood estimator for bearing-only target localization," IEEE Transactions on Signal Processing, vol. 60, no. 12, pp. 6166–6181, 2012.

[2] S. Koteswara Rao, "Algorithm for detection of manoeuvring targets in bearings-only passive target tracking," IEEE Proceedings, vol. 146, no. 3, pp. 141–146, 1999.

[3] S. Koteswara Rao, "Pseudo-linear estimator for bearings-only passive target tracking," IEEE Proceedings, vol. 148, no. 1, pp. 16–22, 2001.

[4] F. Zhang, X.-P. Zhou, X.-H. Chen, and R.-L. Liu, "Particle filter for underwater bearings-only passive target tracking," in Proceedings of the Pacific-Asia Workshop on Computational Intelligence and Industrial Application (PACIIA ’08), pp. 847–851, Wuhan, China, December 2008.

[5] J. Qian, X. Jin, and Z. Yu, "Energy-efficient Node Selection for acoustic source localization in wireless sensor network," in Proceedings of the 6th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM ’10), pp. 1–4, September 2010.

[6] L. M. Kaplan, "Global node selection for localization in a distributed sensor network," IEEE Transactions on Aerospace and Electronic Systems, vol. 42, no. 1, pp. 113–135, 2006.

[7] S. Zhang, J. Wang, X. Liu, and J. Cao, "Range-free selective multilateration for an anisotropic wireless sensor networks," in Proceedings of the 9th Annual IEEE Communication Society Conference on Sensor, Mesh and Ad hoc Communications and Networks, pp. 299–307, 2012.

[8] M. Zoghi and M. H. Kahaei, "Adaptive sensor selection in wireless sensor networks for target tracking," IET Signal Processing, vol. 4, no. 5, pp. 530–536, 2010.

[9] L. M. Kaplan, "Local node selection for localization in a distributed sensor network," IEEE Transactions on Aerospace and Electronic Systems, vol. 42, no. 1, pp. 136–146, 2006.

[10] V. Isler and M. Magdon-Ismail, "Sensor selection in arbitrary dimensions," IEEE Transactions on Automation Science and Engineering, vol. 5, no. 4, pp. 651–660, 2008.

[11] Y. Wang and D. Wang, "Energy-efficient node selection for target tracking in wireless sensor networks," International Journal of Distributed Sensor Networks, vol. 2013, Article ID 830950, 6 pages, 2013.

[12] N. U. Hasan, W. Ejaz, S. Lee, and H. S. Kim, "Knapsack-based energy-efficient node selection scheme for cooperative spectrum sensing in cognitive radio sensor networks," IET Communications, vol. 6, no. 17, pp. 2998–3005, 2012.

[13] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," IEEE Signal Processing Magazine, vol. 19, no. 2, pp. 61–72, 2002.

[14] H. Wang, G. Pottie, K. Yao, and D. Estrin, "Entropy-based sensor selection heuristic for target localization," in Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks, pp. 36–45, ACM, New York, NY, USA, April 2004.
[15] Y. Liu, B. Xu, S. Li, and L. Feng, “Dynamic-group scheduling scheme for maneuvering target tracking in wireless sensor networks,” in *Proceedings of the 8th IEEE International Conference on Control and Automation (ICCA ’10)*, pp. 1300–1304, Xiamen, China, June 2010.

[16] W. Zhao, Y. Han, H. Wu, and L. Zhang, “Weighted distance based sensor selection for target tracking in wireless sensor networks,” *IEEE Signal Processing Letters*, vol. 16, no. 8, pp. 647–650, 2009.

[17] X. Zhen and R. Liyang, “A self-organizing bearings-only target tracking algorithm in wireless sensor network,” *Chinese Journal of Aeronautics*, vol. 22, no. 6, pp. 627–636, 2009.

[18] Y. Han and W. Zhao, “A novel node selection method of bearings-only sensors for target tracking in Wireless sensor Networks,” in *Proceedings of the WRI International Conference on Communications and Mobile Computing (CMC ’09)*, pp. 136–140, January 2009.

[19] F. R. Armaghani, I. Gondal, and J. Kamruzzaman, “Dynamic sensor selection for target tracking in wireless sensor networks,” in *Proceedings of the IEEE Vehicular Technology Conference*, pp. 1–6, September 2011.

[20] H. Wang, K. Yao, and D. Estrin, “Information-theoretic approaches for sensor selection and placement in sensor networks for target localization and tracking,” *Journal of Communications and Networks*, vol. 7, no. 4, pp. 438–449, 2005.

[21] L. M. Kaplan, “Transmission range control during autonomous node selection for wireless sensor networks,” in *Proceedings of the IEEE Aerospace Conference Proceedings*, pp. 2072–2087, March 2004.

[22] H. Ma, “Sensing nodes selection scheme for distributive target tracking in wireless sensor networks,” in *Proceedings of the 37th Annual Conference of the IEEE Industrial Electronics Society (IECON ’11)*, pp. 2164–2169, November 2011.

[23] F. Zhao, J. Liu, J. Liu, L. Guibas, and J. Reich, “Collaborative signal and information processing: an information-directed approach,” *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1199–1209, 2003.

[24] L. M. Kaplan, “Node selection for target tracking using bearing measurements from unattended ground sensors,” in *Proceedings of the Aerospace Conference*, pp. 2137–2152, 2003.

[25] E. Masazade, R. Niu, and P. K. Varshney, “An approximate dynamic programming based non-myopic sensor selection method for target tracking,” in *Proceedings of the 46th IEEE Annual Conference on Information Sciences and Systems (CISS ’12)*, pp. 1–6, 2012.

[26] J. C. Chen, R. E. Hudson, and K. Yao, “Maximum-likelihood source localization and unknown sensor location estimation for wideband signals in the near-field,” *IEEE Transactions on Signal Processing*, vol. 50, no. 8, pp. 1843–1854, 2002.

[27] A. G. O. Mutambara, *Decentralized Estimation and Control for Multisensor Systems*, CRC Press, Boca Raton, Fla, USA, 1998.

[28] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[29] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.
