IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 11, 2018, accepted October 31, 2018, date of publication November 9, 2018, date of current version December 3, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2879536

An Adaptive Sampling Algorithm for Target Tracking in Underwater Wireless Sensor Networks

YANLONG SUN[1], YAZHOU YUAN[1], XIAOLEI LI[1], QIMIN XU[2], AND XINPING GUAN[2], (Fellow, IEEE)
[1]Institute of Electrical Engineering, Yanshan University, Qinhuangdao 066004, China
[2]Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China

Corresponding author: Yazhou Yuan (yzyuan@ysu.edu.cn)

This work was supported in part by the National Nature Science Foundation of China under Grant 61803328, in part by the Natural Science Foundation of Hebei Province under Grant F201703084, and in part by the Post-Doctoral Priority Funding of Hebei Province under Grant B2017003009.

ABSTRACT Target tracking is an important application of underwater wireless sensor networks (UWSNs). Due to the energy constraint and energy imbalanced dissipation of underwater nodes, it is a challenge to maximize the energy efficiency and balance energy consumption simultaneously. In this paper, we propose an adaptive sampling algorithm for target tracking in UWSNs to address this issue. First, for maximizing the energy efficiency, we design an adaptive sampling interval adjustment (ASIA) method using a two-input-single-output fuzzy logic controller. In this method, the sampling interval is adaptively adjusted to make the actually uncertainty equal to the uncertainty threshold, which minimizes the sampling frequency and then reduces the energy consumption of information exchange. Second, for balancing the energy consumption, we develop a dynamic uncertainty threshold adjustment (DUTA) method using a single-input-single-output fuzzy logic controller. According to the residual energy of network nodes, the DUTA method dynamically adjusts the uncertainty threshold in the ASIA method, which changes the sampling frequency for avoiding premature death of nodes. Finally, the simulations show that, compared to the existing adaptive sampling algorithm, the proposed algorithm not only saves about 36% of energy but also alleviates the imbalance of energy consumption in different parts of the tracking area.

INDEX TERMS Underwater wireless sensor networks (UWSNs), target tracking, adaptive sampling, fuzzy logic controller.

I. INTRODUCTION

In the past few decades, underwater wireless sensor networks (UWSNs) have attracted more and more attention in military and civilian fields [1]. UWSNs have many advantages such as self-organization, continuous operation, and wide coverage [2]. Due to these advantages, UWSNs provide a promising solution to target tracking in large three-dimensional underwater area [3]. In UWSNs, nodes are mostly powered by batteries, and it is difficult to replace or recharge batteries in underwater environments [4]–[6]. What's worse, underwater acoustic channel characteristics, such as limited bandwidth, large propagation delay, and high bit error rate, result in higher energy consumption of underwater acoustic communication than terrestrial radio communication [7]–[9]. Therefore, designing more energy-efficient algorithms for target tracking in UWSNs is critical and challenging to prolong the lifetime of nodes.

Since increasing the sampling interval during target tracking can reduce the sampling frequency and then reduce the energy consumption of information exchange between underwater nodes, some efforts have been made in the adaptive sampling algorithm to improve energy efficiency. Kose and Masazade [10] proposed an adaptive sampling algorithm that determined the sampling interval by minimizing the predicted uncertainty of the estimated target position at the next sample moment. Compared to the algorithm that sampled a target using a fixed minimum sampling interval, this algorithm reduced the sampling frequency and energy consumption while providing similar estimation performance. Some works [10]–[15] determined the sampling interval by a value larger than the minimum predicted uncertainty, in order to further reduce the sampling frequency and energy consumption. Kose and Masazade [10] proposed another algorithm which determined the sampling interval based on the 110%

value of the minimum predicted uncertainty. This method significantly reduced the sampling frequency and energy consumption, but the uncertainty was sometimes too large or too small. Too large uncertainty made the user unable to accept the estimated target position, too small uncertainty led to too little energy savings. Xiao *et al.* [11]–[13] set an uncertainty threshold and selected the sampling interval by making the predicted uncertainty at the next sampling moment as large as possible without exceeding the uncertainty threshold. This algorithm not only reduced energy consumption but also avoided the problem of too large or too small uncertainty. However, the sampling interval of this algorithm was only selected from a limited number of candidate values. So Lin *et al.* [14], [15] proposed an improved algorithm which determined the sampling interval by solving an equation that the predicted uncertainty at the next sampling moment equalled to the uncertainty threshold. This algorithm further reduced the sampling frequency and energy consumption without exceeding the uncertainty threshold.

However, there are two shortcomings in [15]. Firstly, without exceeding the uncertainty threshold, the energy saved is the most when the actually obtained uncertainty is equal to the uncertainty threshold. Since the predicted uncertainty is inaccurate, the actually obtained uncertainty is less than the uncertainty threshold, that is, the energy efficiency has not yet reached the maximum. Secondly, increasing the uncertainty threshold can reduce the sampling frequency and then reduce the energy consumption of information exchange between underwater nodes. In UWSNs, the energy consumption in different parts of the tracking area is imbalanced [16], which will lead to premature death of some nodes and shorten the lifetime of networks [17]. The algorithm in [15] does not consider using a dynamic uncertainty threshold to change the sampling frequency for balancing the energy consumption.

To overcome these two shortcomings, in this paper, we propose an adaptive sampling algorithm for target tracking in UWSNs. Firstly, instead of using the predicted uncertainty, we design an adaptive sampling interval adjustment (ASIA) method using a two-input-single-output fuzzy logic controller. In this method, the sampling interval is adaptively adjusted to make the actually uncertainty equal to the uncertainty threshold, which minimizes the sampling frequency and then reduces the energy consumption of information exchange. Secondly, Instead of using a fixed uncertainty threshold, we develop a dynamic uncertainty threshold adjustment (DUTA) method using a single-input-single-output fuzzy logic controller. According to the residual energy of network nodes, the DUTA method dynamically adjusts the uncertainty threshold in the ASIA method, which changes the sampling frequency for avoiding premature death of nodes. Finally, simulations have been carried out to verify the correctness of the analysis and the effectiveness of the proposed algorithm.

In summary, the main contributions of this paper include:

- We design an adaptive sampling interval adjustment (ASIA) method using a two-input-single-output fuzzy logic controller to adaptively adjust the sampling interval. This method achieves maximum energy efficiency without exceeding the uncertainty threshold.
- We develop a dynamic uncertainty threshold adjustment (DUTA) method using a single-input-single-output fuzzy logic controller to dynamically adjust the uncertainty threshold in the ASIA method. This method alleviates the imbalance of energy consumption in different parts of the tracking area.

The remainder of this paper is organized as follows: Section II describes the system model of target tracking in UWSNs. Section III describes the details of the proposed algorithm. Section IV evaluates the performance of the proposed algorithm through a simulation experiment. Section V concludes the paper.

II. SYSTEM MODEL OF TARGET TRACKING IN UWSNs

A. ARCHITECTURE OF UWSNs

Fig. 1 shows a reference architecture for target tracking in UWSNs. There are three types of nodes in the networks according to the deployment position [18]–[20].

- Surface nodes are deployed on the surface of the water. Each surface node is equipped with an acoustic transceiver to communicate with other nodes in the water, and equipped with a radio transmitter to communicate with other surface nodes and the command center.
- Bottom nodes are anchored to the bottom of the water. Each bottom node uses an acoustic transceiver to interconnect with other nodes, and uses optical cables to interconnect with the command center.
- Submarine nodes are deployed in underwater via AUVs, ROVs, or gliders. Nodes can adjust their position in order to form a three-dimensional tracking area. Nodes also are equipped with acoustic transceivers.

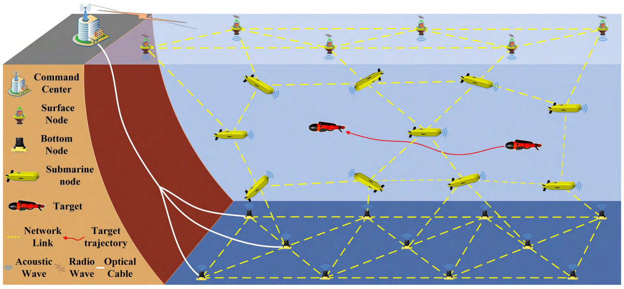


FIGURE 1. The architecture of underwater wireless sensor networks.

The above three types of nodes are constructed into a network through single-hop or multi-hop acoustic communication, cooperate with each other for target tracking, and communicate with the command center through optical cable or radio.

B. DYNAMIC CLUSTERING METHOD

To obtain target data with high energy efficiency, dynamic clustering methods are usually used for target tracking in

UWSNs. In those methods, several nodes are selected to form a cluster at each step of the target tracking. Each cluster consists of a cluster head (CH) node and several cluster member (CM) nodes. The CH node is responsible for managing the intra-cluster and inter-cluster collaboration, while CM nodes are responsible for obtaining target data and sending data to the CH node. In this paper, a dynamic clustering method based on the nearest neighborhood collaboration is used [21].

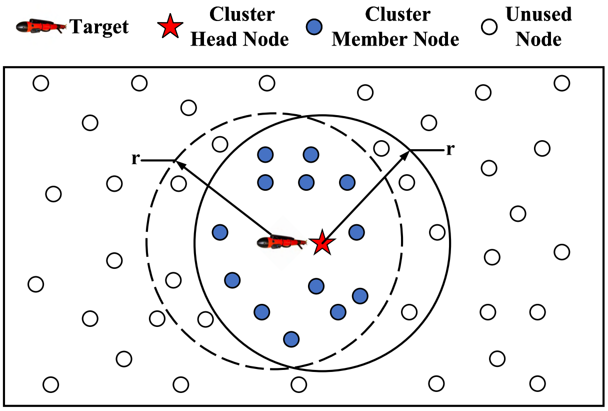


FIGURE 2. The process of forming a cluster based on the nearest neighborhood collaboration.

Fig. 2 shows the process of forming a cluster based on the nearest neighborhood collaboration. The red-black fish-shaped object represents the estimated position of the target. The pentagram-shaped object is the closest node to the estimated position of the target, and is selected as the CH node. Assuming that the detection distance of all nodes is r. All nodes in the dashed circle are the neighbor nodes of the estimated target position, and all the nodes in the solid circle are the neighbor nodes of the CH node. The CH node selects nodes in the intersection of the two circles as CM nodes, which are represented by solid blue circles. Nodes that are not used at the current step are represented by black circles.

C. DISTANCE MEASUREMENT METHOD

After forming a cluster, distances between cluster member nodes and the target need to be measured. In this paper, the 3DUT algorithm is selected as the distance measurement method because it does not require time synchronization [16].

Fig. 3 shows the process of measuring the distance between a cluster member node and the target. The cluster head node sends a ping to the target at the time t_{CH}, and receives the echo at the time t'_{CH}. The distance between the cluster head node and the target d_1 can be calculated by

$$d_1 = C \cdot (t'_{CH} - t_{CH})/2, \tag{1}$$

where C is the underwater sound speed. After the cluster member node receives the echo from the target, the time difference between the arrival of the ping and the echo Δt can be calculated by

$$\Delta t = (d_1 + d_2)/C - d_3/C, \tag{2}$$

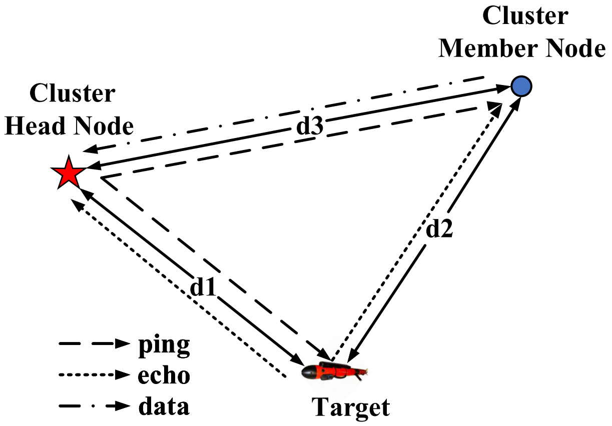


FIGURE 3. The process of measuring the distance between a cluster member node and the target.

where d_2 is the distance between the cluster member node and the target, d_3 is the distance between the cluster member node and the cluster head node. d_3 can be calculated through the operation of the cluster head coordinates and the cluster member coordinates. After the cluster head node receives Δt from the cluster member node, d_2 can be calculated by

$$d_2 = \Delta t \cdot C + d_3 - d_1. \tag{3}$$

D. TARGET POSITION ESTIMATION METHOD

After obtaining all distances between cluster member nodes and the target, the cluster head node can calculate the position of the target by some methods. As shown in Fig. 4(a), when there is no distance measurement error, the trilateration method can determine the intersection of three circles [22]. The centers of these three circles are the coordinates of the cluster member nodes, the radii are the distances from the cluster member nodes to the target, and the intersection is the target position. Because the distance measurement error is inevitable, these circles may not intersect at one point, as shown in Fig. 4(b). To solve this problem, the least squares method is used to find a best estimated position of the target by minimizing the sum of the squares of the residuals of cluster member nodes from the estimated target position [23]. However, the least squares method does not consider the

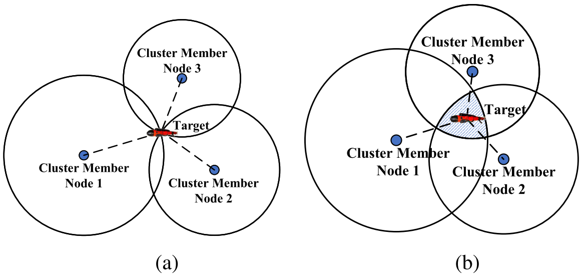


FIGURE 4. Determining the target position. (a) Determining the target position when the measured distances have no error. (b) Determining the target position when the measured distances have errors.

correlation between the previous and current target data, does not know any statistical information related to the estimated quantity, and the result of its positioning solution is poor in robustness. In order to improve the accuracy and stability of the position estimation, the extended kalman filter (EKF) algorithm is adopted.

Define target's process state vector at the $k-$th step as

$$X(k) = (x(k), \dot{x}(k), y(k), \dot{y}(k), z(k), \dot{z}(k))^T, \tag{4}$$

where $x(k)$, $y(k)$ and $z(k)$ are the x-, y- and z-coordinates of the target, $\dot{x}(k)$, $\dot{y}(k)$ and $\dot{z}(k)$ are the x-, y- and z-velocities of the target. In this paper, we only consider a single target tracking problem, and a constant velocity target motion model [24] with state equation

$$X(k) = F(T(k-1))X(k-1) + v(k-1), \tag{5}$$

where $T(k-1)$ is the sampling interval between the $(k-1)-$th step and the $k-$ th step, $F(T(k-1))$ is the state transition matrix, $v(k-1)$ is the process noise which is assumed to be zero mean gaussian noise with process noise covariance matrix $Q(T(k-1))$. The $F(T(k-1))$ can be expressed by

$$F = \begin{bmatrix} 1 & T(k-1) & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T(k-1) & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T(k-1) \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{6}$$

The $Q(T(k-1))$ can be expressed by

$$Q(T(k-1)) = q \cdot \begin{bmatrix} Q_{sub} & 0 & 0 \\ 0 & Q_{sub} & 0 \\ 0 & 0 & Q_{sub} \end{bmatrix}, \tag{7}$$

where q is the intensity of the process noise, Q_{sub} is given by

$$Q_{sub} = \begin{bmatrix} T(k-1)^3/3 & T(k-1)^2/2 \\ T(k-1)^2/2 & T(k-1) \end{bmatrix}. \tag{8}$$

Suppose there are n cluster member nodes involved in target tracking at the $k-$th step $CM_1(k)$, ..., $CM_n(k)$, which give the measurements of the target $z_1(k)$, ..., $z_n(k)$. The measurement equation for the entire cluster is as

$$\begin{aligned} Z(k) = \begin{bmatrix} z_1(k) \\ z_2(k) \\ \vdots \\ z_n(k) \end{bmatrix} &= \begin{bmatrix} h_1(X(k)) \\ h_2(X(k)) \\ \vdots \\ h_n(X(k) \end{bmatrix} + \begin{bmatrix} w_1(k) \\ w_2(k) \\ \vdots \\ w_n(k) \end{bmatrix} \\ &= h(X(k)) + w(k). \end{aligned} \tag{9}$$

where $h_i(X(k))$ is the measurement function of $CM_i(k)$ $(i = 1, 2, ...n)$, $w_i(k)$ is the uncorrelated measurement gauss white noise of $CM_i(k)$ with covariance $R_i(k)$, $h(X(k))$ is the vector form of the measurement functions, $w(k)$ is the vector form of the measurement noises with covariance matrix $R(k)$. The process and measurement noise are assumed to be mutually independent. $R(k)$ can be expressed by

$$R(k) = diag\left[R_1(k), \quad R_2(k), \quad \cdots \quad, R_n(k)\right]. \tag{10}$$

The $h_i(X(k))$ can be expressed by

$$\begin{aligned} h_i(X(k)) = ((x(k) - x_{CM_i}(k))^2 + (y(k) - y_{CM_i}(k))^2 \\ + (z(k) - z_{CM_i}(k))^2)^{1/2} \end{aligned} \tag{11}$$

Where $(x_{CM_i}(k), y_{CM_i}(k), z_{CM_i}(k))$ denotes the position of $CM_i(k)$. $h(X(k))$ have the jacobian matrix $H(k)$

$$\begin{aligned} &H(k) \\ &= \begin{bmatrix} \frac{x(k)-x_{CM_1}(k)}{h_1(X(k))} & 0 & \frac{y(k)-y_{CM_1}(k)}{h_1(X(k))} & 0 & \frac{z(k)-z_{CM_1}(k)}{h_1(X(k))} & 0 \\ \frac{x(k)-x_{CM_2}(k)}{h_2(X(k))} & 0 & \frac{y(k)-y_{CM_2}(k)}{h_2(X(k))} & 0 & \frac{z(k)-z_{CM_2}(k)}{h_2(X(k))} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{x(k)-x_{CM_n}(k)}{h_n(X(k))} & 0 & \frac{y(k)-y_{CM_n}(k)}{h_n(X(k))} & 0 & \frac{z(k)-z_{CM_n}(k)}{h_n\ (X(k))} & 0 \end{bmatrix}. \end{aligned} \tag{12}$$

EKF is a recursive estimator, and can be divided into predict phase and update stage [25]. The predict phase uses the estimated state of the previous step to predict an estimate state of the current step. The update phase combines the current prediction with current measurement data to refine the estimated state. Equations for predict phase and update stage are as follows.

Predict:

$$\hat{X}(k|k-1) = F(T(k-1))\hat{X}(k-1|k-1) \tag{13}$$

$$\begin{aligned} P(k|k-1) = F(T(k-1))P(k-1|k-1)F(T(k-1))^T \\ + Q(T(k-1)) \end{aligned} \tag{14}$$

where

$\hat{X}(k|k-1)$ Predicted state estimate,
$P(k|k-1)$ Predicted covariance estimate.

Update:

$$\tilde{y}(k) = Z(k) - h(\hat{X}(k|k-1)) \tag{15}$$

$$S(k) = H(k)P(k|k-1)H^T(k) + R(k) \tag{16}$$

$$K(k) = P(k|k-1)H(k)^T S(k)^{-1} \tag{17}$$

$$\hat{X}(k|k) = \hat{X}(k|k-1) + K(k)\tilde{y}(k) \tag{18}$$

$$P(k|k) = (I - K(k)H(k))P(k|k-1) \tag{19}$$

where

$\tilde{y}(k)$ Innovation residual,
$S(k)$ Innovation covariance,
$K(k)$ Kalman gain,
$\hat{X}(k|k)$ Updated state estimate,
$P(k|k)$ Updated covariance estimate.

After the above EKF operations are used at the $k-$th step, the cluster head node obtains the estimated position of the target $(\hat{x}(k|k), \hat{y}(k|k), \hat{z}(k|k))$, and the measure of the estimated accuracy of the state estimate $P(k|k)$.

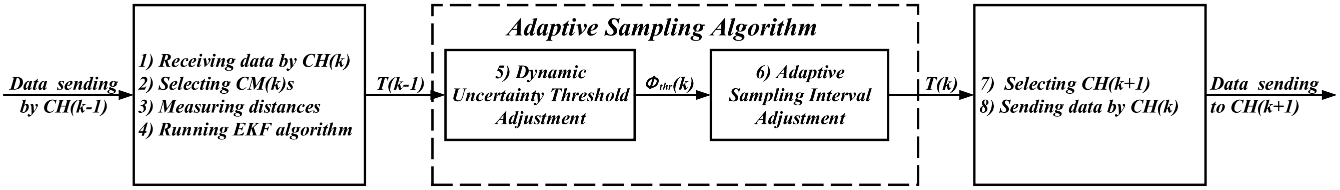


FIGURE 5. The process of target tracking using the adaptive sampling algorithm at the $k-$th step.

III. DETAILED DESCRIPTION OF THE ADAPTIVE SAMPLING ALGORITHM FOR TARGET TRACKING IN UWSNs

A. THE PROCESS OF TARGET TRACKING USING THE ADAPTIVE SAMPLING ALGORITHM

Target tracking is a recursive process consisting of several steps, and each step performs the same operation. So this subsection only takes the operation of the $k-$th step as an example. As shown in Fig. 5, at the $k-$ th step, the process of target tracking using the adaptive sampling algorithm includes the following operations:

1) $CH(k)$ receives data from $CH(k-1)$ to complete the cluster head switch.
2) $CH(k)$ selects $CM(k)s$ using the method given in Subsection II-B to form a cluster.
3) $CH(k)$ obtains distances between the $CM(k)s$ and the target using the method presented in Subsection II-C.
4) $CH(k)$ runs the EKF algorithm to get the target state estimation $\hat{X}(k|k)$ and the covariance matrix $P(k|k)$ provided in Subsection II-D
5) $CH(k)$ uses the dynamic uncertainty threshold adjustment method explained in Subsection III-D to calculate the uncertainty threshold at the $k-$th step $\Phi_{thr}(k)$.
6) $CH(k)$ uses the adaptive sampling interval adjustment method explained in Subsection III-C to yield $T(k)$ that the sampling interval between the $k-$th step and the $(k+1)-$th step.
7) $CH(k)$ selects the $CH(k+1)$ uses the method given in Subsection II-B.
8) $CH(k)$ sends data to $CH(k+1)$ to complete the cluster head switch.

B. TRACKING ACCURACY AND ENERGY CONSUMPTION

1) TRACKING ACCURACY

The estimated position error and the uncertainty of the estimated target position are often used to stand for the tracking accuracy. The estimated position error, which is defined as the distance between the true position of the target $(x(k), y(k), z(k))$ and the estimated target position $(\hat{x}(k|k), \hat{y}(k|k), \hat{z}(k|k))$ [23], can be expressed by

$$\Psi(k) = ((x(k)-\hat{x}(k|k))^2 + (y(k)-\hat{y}(k|k))^2 + (z(k)-\hat{z}(k|k))^2)^{1/2}. \quad (20)$$

However, the true position of the target is not available in the actual target tracking, the cluster header node only can get the target state estimation $\hat{X}(k|k)$ and the covariance matrix $P(k|k)$, which is the measure of the estimated accuracy of the target state. In this paper, the uncertainty of the estimated target position $\Phi(k)$, which is defined as the trace of the position covariance matrix, is used to stand for the tracking accuracy [26]. The state covariance matrix $P(k|k)$ can be expressed as

$$P(k|k) = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} & \sigma_{15} & \sigma_{16} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} & \sigma_{24} & \sigma_{25} & \sigma_{26} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} & \sigma_{34} & \sigma_{35} & \sigma_{36} \\ \sigma_{41} & \sigma_{42} & \sigma_{43} & \sigma_{44} & \sigma_{45} & \sigma_{46} \\ \sigma_{51} & \sigma_{52} & \sigma_{53} & \sigma_{54} & \sigma_{55} & \sigma_{56} \\ \sigma_{61} & \sigma_{62} & \sigma_{63} & \sigma_{64} & \sigma_{65} & \sigma_{66} \end{bmatrix}. \quad (21)$$

According to (4), the associated position covariance matrix $\Sigma(k|k)$ can be represented by

$$\Sigma(k|k) = \begin{bmatrix} \sigma_{11} & \sigma_{13} & \sigma_{15} \\ \sigma_{31} & \sigma_{33} & \sigma_{35} \\ \sigma_{51} & \sigma_{53} & \sigma_{55} \end{bmatrix}. \quad (22)$$

The uncertainty of the estimated target position $\Phi(k)$ can be represented by

$$\Phi(k) = trace(\Sigma(k|k)) = \sigma_{11} + \sigma_{33} + \sigma_{55}, \quad (23)$$

where σ_{11}, σ_{33} and σ_{55} are the uncertainty of the estimated target position in the x-, y- and z-coordinates.

Define $\Phi_{thr}(k)$ as the uncertainty threshold at the $k-$th step. Tracking accuracy at the $k-$th step is considered satisfactory if

$$\Phi(k) \leq \Phi_{thr}(k). \quad (24)$$

Otherwise, it is considered to be unsatisfactory.

2) ENERGY CONSUMPTION

According to the energy model [27], to transmit b bits data from one sensor node to another over a distance d, the energy consumption is

$$E_{com}(b, d) = E_t(b, d) + E_r(b), \quad (25)$$
$$E_t(b, d) = E_{te} \times b + E_{amp}(b, d), \quad (26)$$
$$E_r(b) = E_{re} \times b, \quad (27)$$

where $E_t(b, d)$ and $E_r(b)$ are the energy consumed by the transmitter and receiver, E_{te} and E_{re} are the unit energy consumed by the electronics of the transmitter and receiver to process one bit of data, $E_{amp}(b, d)$ is energy consumed by the power amplifier to ensure the transmission of underwater acoustic signals. $E_{amp}(b, d)$ can be expressed as

$$E_{amp}(b, d) = P_t(d) \times \frac{b}{v}, \tag{28}$$

where $P_t(d)$ is the transmission power which a function of the distance d, v represents the data transmission speed. In UWSNs, the acoustic signals propagate in a spherical fashion [28]. $P_t(d)$ is related to the transmission power intensity at $1m$ from the source I_0 and the source level SL [29]

$$P_t(d) = 4\pi \times 1^2 \times I_0, \tag{29}$$

$$SL = 10 \log \frac{I_0}{I_{ref}}, \tag{30}$$

where I_{ref} is $0.67 \times 10^{-18} \mu Pa$. According to the passive sonar equation [30], the minimum source level SL is

$$SL - TL - NL + DI = DT, \tag{31}$$

where TL, NL DI and DT are transmission loss, noise level, directivity index and detection threshold respectively. TL can be expressed by [31]

$$TL = 20 \log d + \alpha d \times 10^{-3} + A, \tag{32}$$

where α is the absorption coefficient, A is the transmission loss anomaly that accounts for multipath propagation, refraction and other phenomenon. According to (29), (30), (31), and (32), $P_t(d)$ can be expressed by

$$P_t(d) = 4\pi \times 10^{\frac{20 \log d + \alpha d \times 10^{-3} + A + NL - DI + DT}{10}} \times 0.67 \times 10^{-18} \tag{33}$$

Suppose there are n cluster members at the $k-$th step $CM_1(k)$, ..., $CM_n(k)$, and their distances from the cluster head is $d_{CM_1(k)}$, ..., $d_{CM_n(k)}$. The distance between the cluster head $CH(k)$ and the estimated position of the target is l_1. $CM_i(k)$ ($i \in \ 1, 2, ...n$) has the largest distance from the estimated position of the target l_2, $CM_j(k)$ ($j \in \ 1, 2, ...n$) has the largest distance from the cluster head l_3. Based on the energy model and the process of target tracking previously given, the energy consumption of information exchange at the $k-$th step can be expressed by

$$E_{total}(k) = E_{sr}(k) + E_{se}(k) + E_{me}(k) + E_{st}(k), \tag{34}$$

$$E_{sr}(k) = E_r(b_{sr}), \tag{35}$$

$$E_{se}(k) = E_t(b_{se}, l_3) + n \cdot E_r(b_{se}) + \sum_{i=1}^{n} E_t(b_{se}, d_{CM_i(k)}) + n \cdot E_r(b_{se}), \tag{36}$$

$$E_{me}(k) = E_t(b_{me}, l_1 + l_2) + (n + n + 1) \cdot E_r(b_{me}) + \sum_{i=1}^{n} E_t(b_{me}, d_{CM_i(k)}) + n \cdot E_r(b_{me}), \tag{37}$$

$$E_{st}(k) = E_t(b_{st}, d_{st}(k)), \tag{38}$$

where $E_{total}(k)$ is the total energy consumed at the $k-$th step, $E_{sr}(k)$ is the energy consumed in receiving data from $CH(k-1)$ to complete the cluster head switch, $E_{se}(k)$ is the energy consumed in selecting cluster members, $E_{me}(k)$ is the energy consumed in measuring distances, $E_{st}(k)$ is the energy consumed in transmitting data to $CH(k+1)$ to complete the cluster head switch, b_{sr}, b_{se}, b_{me}, b_{st} are the number of bits in the data packet used in receiving data from $CH(k-1)$, selecting cluster members, measuring distances, transmitting data to $CH(k+1)$, $d_{st}(k)$ is the distance between $CH(k)$ and $CH(k+1)$. In this paper, we ignore the energy consumption by running algorithm and calculating data.

C. ADAPTIVE SAMPLING INTERVAL ADJUSTMENT METHOD

Fig. 6 illustrates the adaptive sampling interval adjustment (ASIA) method based on a two-input-single-output fuzzy logic controller. The uncertainty of the estimated target position $\Phi(k)$ can be calculated according to (23), and the uncertainty threshold $\Phi_{thr}(k)$ can be obtained according to the description of Subsection III-D. Define $E(k)$, the error between the uncertainty threshold and the uncertainty at the $k-$th step, as

$$E(k) = \Phi_{thr}(k) - \Phi(k). \tag{39}$$

Define $EC(k)$, the error change ratio in the sampling interval $T(k-1)$, as

$$EC(k) = \frac{E(k) - E(k-1)}{T(k-1)}. \tag{40}$$

According to the input $E(k)$ and $EC(k)$, the output $TC(k)$, which corresponds to the amount of time to increment or decrement the current sampling interval $T(k-1)$, can be obtained by the FLC.

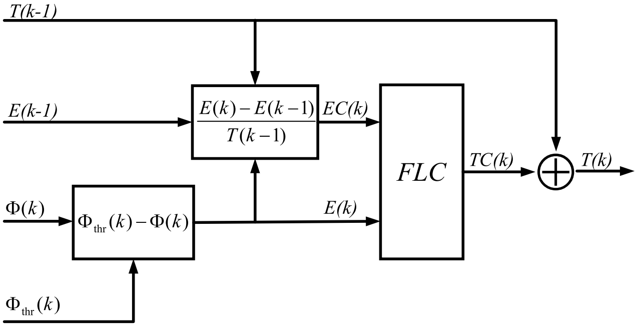


FIGURE 6. The adaptive sampling interval adjustment method based on a two-input-single-output fuzzy logic controller.

As shown in Fig. 7, the implementation process of the FLC comprises of three steps [32]:

1) Fuzzification: The fuzzification process converts crisp inputs to fuzzy sets via input membership functions. As showed in Fig. 8, there are two membership functions for the inputs $E(k)$ and $EC(k)$ and one for the output $TC(k)$. Membership functions are stored in the knowledge base. For $E(k)$ and $EC(k)$, the fuzzy states both are labeled in the linguistic terms of Negative Big (NB), Negative Medium (NM), Negative

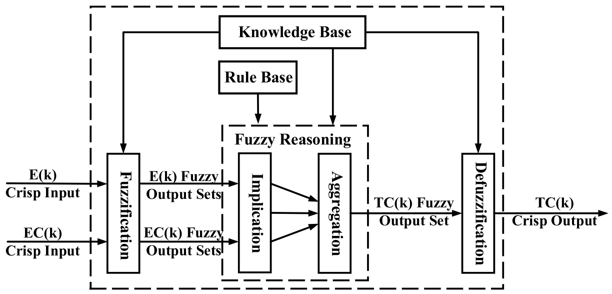


FIGURE 7. The implementation process of the fuzzy logic controller.

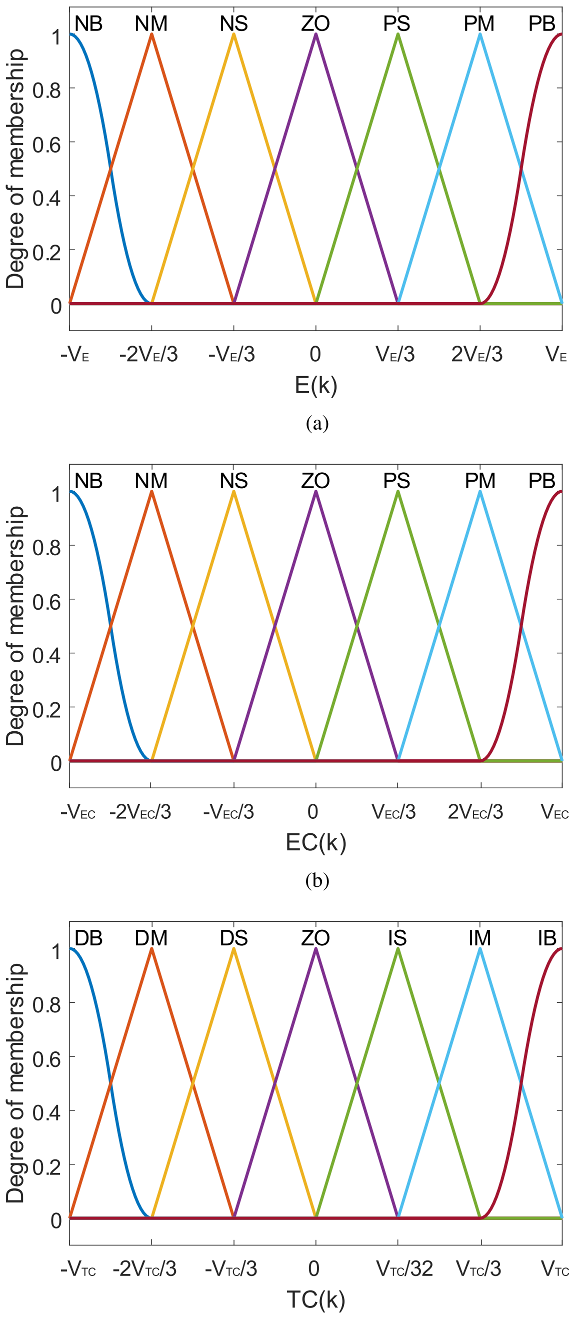
Small (NS), Zero (ZO), Positive Small (PS), Positive Medium (PM), and Positive Big (PB). $TC(k)$ falls into one of seven fuzzy states: Decrease Big (DB), Decrease Medium (DM), Decrease Small (DS), No Change (NC), Increase Small (IS), Increase Medium (IM) and Increase Big (IB). The universe of discourse of $E(k)$, $EC(k)$ and $TC(k)$ are $[-V_E, V_E]$, $[-V_{EC}, V_{EC}]$ and $[-V_{TC}, V_{TC}]$ respectively.

2) Fuzzy Reasoning: After inputs are fuzzified, fuzzy reasoning uses the rule base and knowledge base to map the input values to an output fuzzy set. The rule base is a set of rules based on the experimental data and experiences of the human experts. TABLE. 1 shows the rule base for the adaptive sampling interval adjustment method. There are two steps in the fuzzy reasoning process. The first step, implication, is to find the set of all rules that correspond to the input membership states from the rule base, and to choose the minimum value of the degree of membership of the inputs among the matching rules. The second step, aggregation, is to combine the output fuzzy sets for each rule into a single fuzzy set for output variable by the max method.

3) Defuzzification: To obtain a crisp output value from the aggregated output fuzzy set, defuzzification must be done. In order to get smooth output, the centroid calculation method is selected as the defuzzification method. After defuzzification, the numerical output $TC(k)$ can be obtained.

TABLE 1. Rule base for the adaptive sampling interval adjustment method.

$TC(k)$ $EC(k)$ $E(k)$	NB	NM	NS	ZO	PS	PM	PB
NB	DB	DB	DB	DB	DM	DS	NC
NM	DB	DB	DB	DM	DS	NC	IS
NS	DB	DB	DM	DS	NC	IS	IM
ZO	DB	DM	DS	NC	IS	IM	IB
PS	DM	DS	NC	IS	IM	IB	IB
PM	DS	NC	IS	IM	IB	IB	IB
PB	NC	IS	IM	IB	IB	IB	IB



(a)

(b)

(c)

FIGURE 8. Membership functions for the adaptive sampling interval adjustment method. (a) Membership function of input variable $E(k)$. (b) Membership function of input variable $EC(k)$. (c) Membership function of output variable $TC(k)$.

After the FLC implementation, the addition of the current sampling interval $T(k-1)$ and the sampling interval adjustment amount $TC(k)$ generates a new value for the next sampling interval $T(k)$.

D. DYNAMIC UNCERTAINTY THRESHOLD ADJUSTMENT METHOD

As shown in Fig. 9, the target enters the tracking area from the free area, passes through the outer, middle and inner part of the tracking area in turn, and approaches the protected area. In order to detect the target as soon as it enters into the tracking area, the nodes of the outer part of the tracking area have higher duty cycles and higher energy consumption than the nodes of the inner nodes [16]. To alleviate the imbalance of energy consumption, from the inner part to the outer part, the uncertainty threshold is increased, the sampling frequency and energy consumption are reduced. A single-input-single-output fuzzy logic controller is used to describe this relationship that the closer the target is from the free area, the higher the uncertainty threshold.

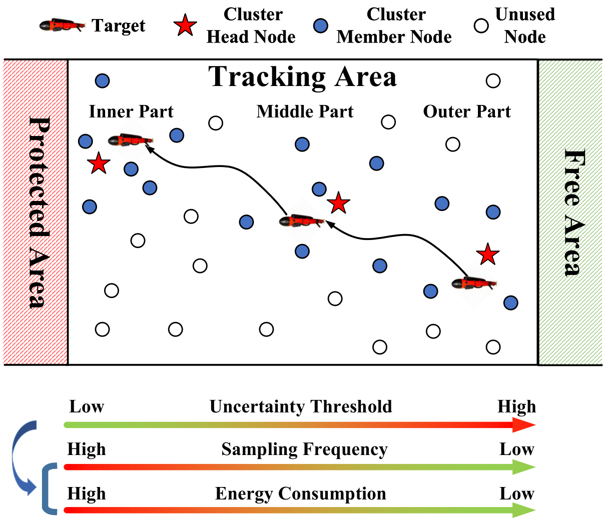


FIGURE 9. The effect of adjusting uncertainty threshold on sampling frequency and energy consumption.

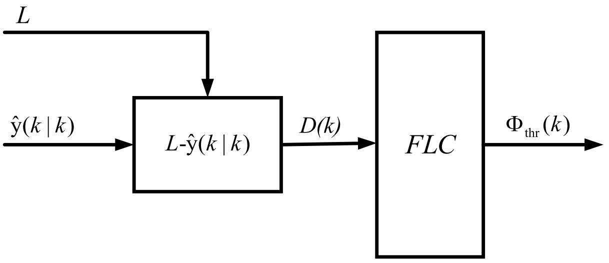
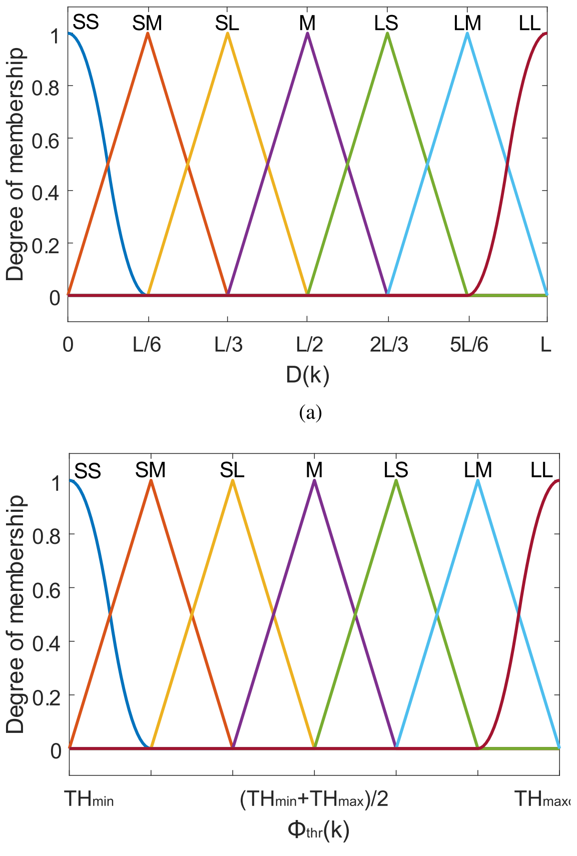


FIGURE 10. The dynamic uncertainty threshold adjustment method based on a single-input-single-output fuzzy logic controller.

Fig. 10 illustrates the dynamic uncertainty threshold adjustment (DUTA) method based on a single-input-single-output fuzzy logic controller. At the $k-$th step, the distance from the target to the protected area $D(k)$ can be defined as

$$D(k) = L - \hat{y}(k|k), \tag{41}$$

where L is the length of the tracking area in the y-direction, $\hat{y}(k|k)$ is the estimated y-coordinate of the target. According to the input $D(k)$, the output $\Phi_{thr}(k)$ can be obtained by the FLC. The implementation process of the FLC is similar to that of the Subsection III-C, but membership functions and the rule base are different. As shown in Fig. 11, both the input and output membership functions use seven fuzzy states,



(a)

(b)

FIGURE 11. Membership functions for the dynamic tracking accuracy threshold adjustment method. (a) Membership function of input variable ***D*(*k*)**.

TABLE 2. Rule base for the dynamic uncertainty threshold adjustment method.

Rule	Antecedent	Consequent
1	**if** $D(k)$ is (SS)	**then** $\Phi_{thr}(k)$ is (SS)
2	**if** $D(k)$ is (SM)	**then** $\Phi_{thr}(k)$ is (SM)
3	**if** $D(k)$ is (SL)	**then** $\Phi_{thr}(k)$ is (SL)
4	**if** $D(k)$ is (M)	**then** $\Phi_{thr}(k)$ is (M)
5	**if** $D(k)$ is (LS)	**then** $\Phi_{thr}(k)$ is (LS)
6	**if** $D(k)$ is (LM)	**then** $\Phi_{thr}(k)$ is (LM)
7	**if** $D(k)$ is (LL)	**then** $\Phi_{thr}(k)$ is (LL)

which are Small-Small (SS), Small-Medium (SM), Small-Large (SL), Medium (M), Large-Small (LS), Large-Medium (LM), and Large-Large (LL). The universe of discourse of $D(k)$ and $\Phi_{thr}(k)$ are $[0, L]$ and $[TH_{min}, TH_{max}]$ respectively. TABLE. 2 shows the rule base for the dynamic uncertainty threshold adjustment method.

IV. SIMULATION EXPERIMENT

In this section, a simulation experiment is carried out to analyze the performance of the proposed algorithm. There are five different sampling algorithms in this experiment. The FLAS(DT) algorithm, which jointly use the ASIA method and the DUTA method, is the adaptive sampling algorithm proposed in this paper. The EEDAS algorithm, which is the adaptive sampling algorithm in [15], is used to compare with the FLAS(DT) to evaluate the performance of the ASIA method. The FLAS(FT) algorithm, which uses the ASIA method and a fixed uncertainty threshold, is used to compare with the FLAS(DT) to evaluate the performance of the DUTA method. The MINS algorithm uses the fixed sampling interval T_{min}. The MAXS algorithm uses the fixed sampling interval T_{max}. The MINS algorithm and MAXS algorithm are fixed sampling algorithms used to compare with the above three adaptive sampling algorithms.

A. SIMULATION SETTINGS

As shown in Fig. 12, the tracking area is a cube of 1000m × 1000m × 1000m. The free area is on the right side of the tracking area, and the protected area is on the left side of the tracking area. They are not presented in the figure. In the tracking area, blue circles stand for the underwater wireless sensor nodes and the number of nodes is 125. All nodes are assumed to be stationary and randomly deployed using a uniform distribution throughout the tracking area and can locate their position with the help of the localization algorithm [33].

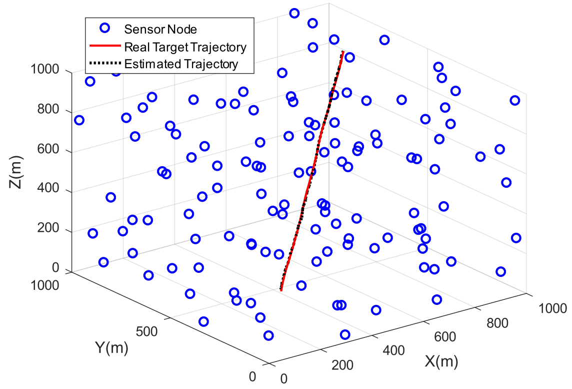


FIGURE 12. Real target trajectory and its estimated trajectory using the proposed algorithm.

As shown in Fig. 12, the target aims to travel from point (200, 200, 200) to point (800, 800, 800) with a constant velocity (5m/s, 5m/s, 5m/s). The intensity of the process noise is q = 0.01. The variance of measurement noise for all nodes is 49. The initial EKF estimation of the target position is (205, 205, 205) and the initial covariance matrix is $9I$, where I is the identity matrix. The fixed uncertainty threshold is 10. The dynamic threshold is bounded within $[TH_{min}, TH_{max}] = [7, 13]$. The sampling interval is bounded within $[T_{min}, T_{max}] = [0.5, 4]$. The membership parameters VE, VEC, and VTC are 3, 2, 2 respectively. For energy consumption model, the following parameters are used: $v = 4kbits/s$, $\alpha = 21dB/km$, $A = 21dB$, $NL = 70dB$, $DI = 0dB$, $DT = 20dB$, $E_{te} = E_{re} = 50nJ/bit$, and $b_{sr} = b_{se} = b_{me} = b_{st} = 256bit$.

In our simulations, the Monte Carlo method is used. We track the target 120 seconds per trial, and the number of trials is $N_{trial} = 100$. Due to the existence of the process noise, the target trajectories in these trials are different.

B. SIMULATION RESULTS

Fig. 12 shows the successful tracking process of the FLAS(DT) algorithm in a randomly selected trial. The estimated trajectory lies very closely to the real target trajectory.

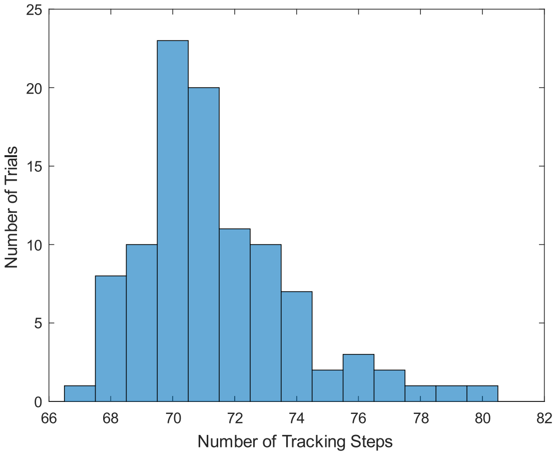


FIGURE 13. Histogram of the number of tracking steps per trial using the proposed algorithm.

Fig. 13 shows the histogram of the number of tracking steps per trial using the FLAS(DT) algorithm. Because of the difference in trajectories, the number of tracking steps per trial varies between 67 and 80.

Fig. 14 shows that the average number of cluster nodes per step of the five algorithms is 15.74, 15.52, 15.43, 15.39 and 15.64 respectively. Although the number of cluster nodes per step is variable, it is much larger than the minimum requirement 4. Redundant cluster nodes at each step reduce the effect of different number of cluster nodes on the uncertainty of the estimated target position. The almost same average number of cluster nodes per step reduces the effect of different number of cluster nodes on the energy consumption.

Fig. 15 shows that the average sampling interval per step of the five algorithms is 0.50s, 1.08s, 1.69s, 1.69s and 4.00s respectively. The larger the sampling interval, the less the number of samples, the less energy is consumed. Due to the effect of the ASIA method, the FLAS(DT)'s average sampling interval per step is almost the same as the FLAS(FT)'s, and is much larger than EEDAS's.

For adaptive sampling algorithms, the sampling interval per step is different, the total number of tracking steps per trial is also different. Therefore, for a fair comparison, we partition the 120 second tracking period of a trial into smaller 6 second time windows, calculate the average uncertainty of the estimated target position and the average estimated position error

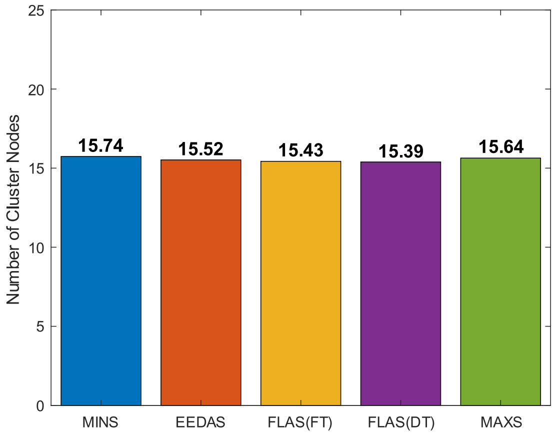


FIGURE 14. The average number of cluster nodes per step of the five algorithms.

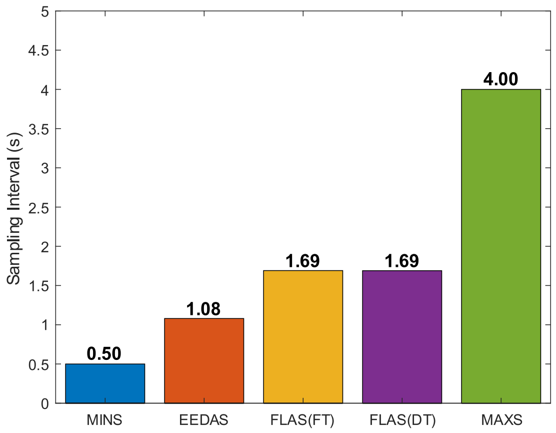


FIGURE 15. The average sampling interval per step of the five algorithms.

in each of these time windows. Let N_{total} be the total number of samples from the target fall in time window $[6(t-1), 6t]$. Define $\Phi_{6s}(t)$ as the average uncertainty of the estimated target position of the tth time window of 100 trials as follows,

$$\Phi_{6s}(t) = \frac{1}{N_{trial}} \sum_{j=1}^{N_{trial}} (\frac{1}{N_{total}} \sum_{m \in [6(t-1),6t]} \Phi_m^j), \qquad (42)$$

where Φ_m^j is the estimated uncertainty at the sampling moment m in the jth trial. Similarly, define $\Psi_{6s}(t)$ as the average estimated position error of the tth time window of 100 trials as follows,

$$\Psi_{6s}(t) = \frac{1}{N_{trial}} \sum_{j=1}^{N_{trial}} (\frac{1}{N_{total}} \sum_{m \in [6(t-1),6t]} \Psi_m^j), \qquad (43)$$

where Ψ_m^j is the estimated position error at the sampling moment m in the jth trial.

The different sampling intervals of the five algorithms significantly affect the respective uncertainty. Fig. 16 shows the uncertainty of the estimated target position of the five algorithms consisting of their respective $\Phi_{6s}(t)$. Since the initial EKF covariance matrix is inaccurate, the estimated uncertainty is very large at the beginning. Due to the effect of EKF, the estimated uncertainty then drops quickly. To balance the uncertainty and energy consumption, the three adaptive sampling algorithms EEDAS, FLAS(FT), and FLAS(DT) all attempt to make the uncertainty of each step equal to the uncertainty threshold. Due to the error in the predicted uncertainty, EEDAS's uncertainty is smaller than the fixed threshold. In contrast, FLAS(DT)'s and FLAS(FT)'s are very closely to the fixed threshold and the dynamic threshold respectively. MAXS's uncertainty is always much larger than the fixed threshold. MINS's uncertainty is always much smaller than the fixed threshold. As shown in Figure 17, the average uncertainty per step of the five algorithms is 4.70, 8.05, 10.72, 10.62 and 16.62 respectively. Due to inaccurate initial estimates, FLAS(FT)'s value 10.72 is slightly higher than the fixed threshold 10. EEDAS's value 8.05 does not reach the fixed threshold 10. Fig. 16 and Fig. 17 can prove that the ASIA method can achieve the maximizes energy efficiency without exceeding the uncertainty threshold, and has better performance than the EEDAS algorithm.

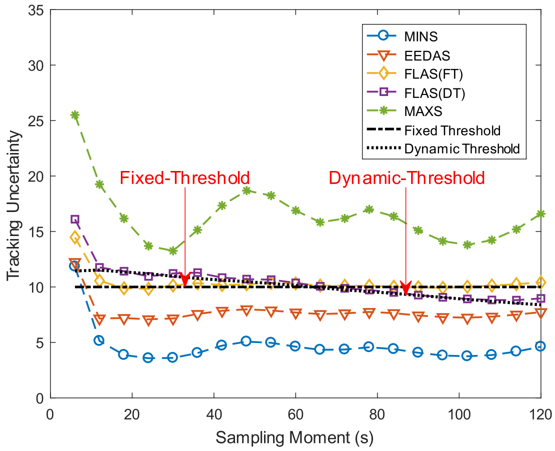


FIGURE 16. The uncertainty of the estimated target position of the five algorithms.

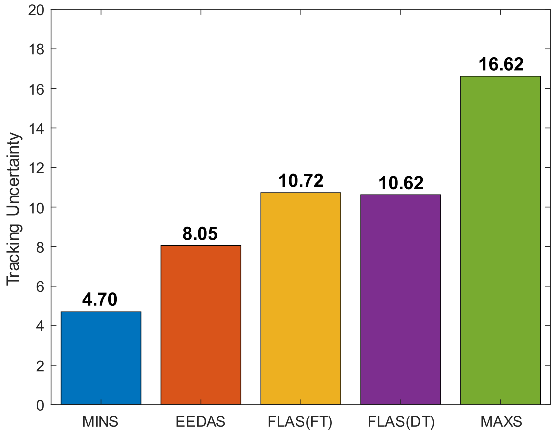


FIGURE 17. The average uncertainty of the estimated target position per step of the five algorithms.

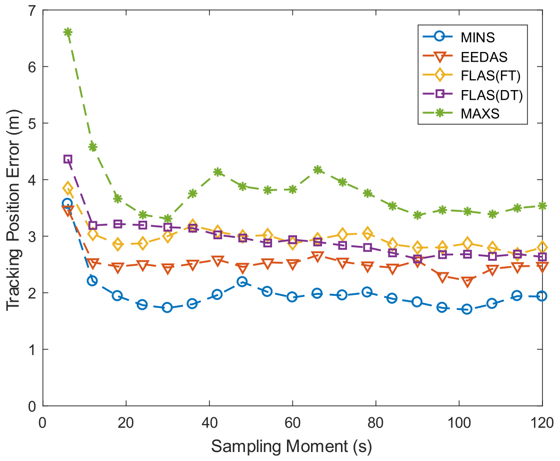


FIGURE 18. The estimated position error of the five algorithms.

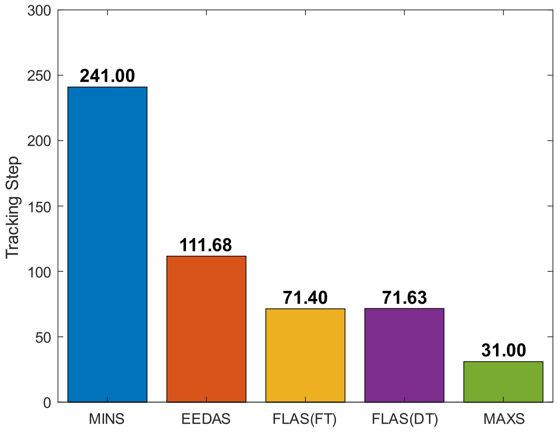


FIGURE 20. The average number of tracking steps per trial of the five algorithms.

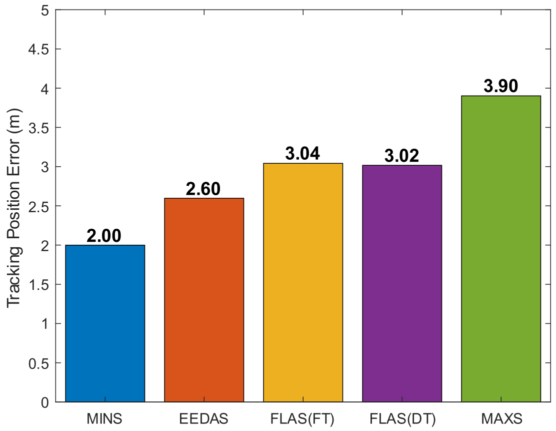


FIGURE 19. The average estimated position error per step of the five algorithms.

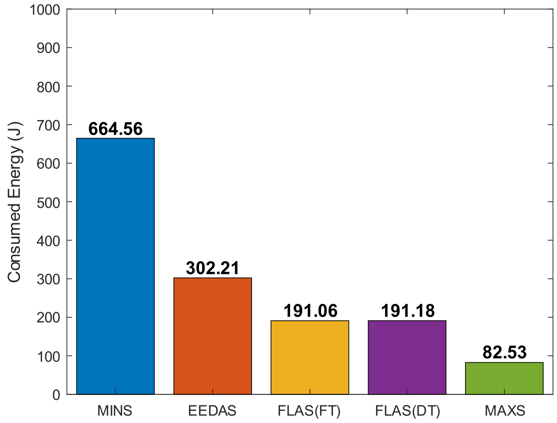


FIGURE 21. The total energy consumption of the five algorithms.

Fig. 18 shows the estimated position error of the five algorithms consisting of their respective $\Psi_{6s}(t)$. The estimated position error of the five algorithms is similar to the uncertainty. FLAS(DT) and FLAS(FT) can control the estimated position error around 3. Since the uncertainty used by the EEDAS is less than the threshold, the estimated position error is smaller than FLAS(DT)'s and FLAS(FT)'s. As shown in Figure 19, the average estimated position error per step of the five algorithms is 2.00, 2.60, 3.04, 3.02 and 3.90 respectively. EEDAS's value 2.60 is smaller than FLAS(DT)'s and FLAS(FT)'s. Although the estimated position error of EEDAS is reduced, this is at the cost of increasing the number of samples.

Since the tracking period per trial is the same, the difference in the sampling interval of the five algorithms causes a difference in the average number of tracking steps per trial. As shown in Fig. 20, the average number of tracking steps per trial of the five algorithms is 241.00, 111.68, 71.40, 71.63, and 31.00 respectively. MINS's number of tracking steps is the largest, and MAXS's number of tracking steps is the smallest. FLAS(DT)'s number of tracking steps is almost the same as FLAS(FT)'s, and much smaller than EEDAS's.

Since the average number of cluster nodes per step of the five algorithms is almost the same, the average energy consumption per step of the five algorithms is also almost the same. In the entire target tracking process, the greater the number of tracking steps, the greater the total energy consumption. As shown in Fig. 21, the total energy consumption of the five algorithms is $664.56J$, $302.21J$, $191.06J$, $191.18J$ and $82.53J$ respectively. Both FLAS(DT) and FLAS(FT) algorithms save about 36% of energy in comparison to the EEDAS algorithm, and save about 71% of energy in comparison to the MINS algorithm. Although the MAXS algorithm consumes the least energy, its estimated position error is too large. By comparing the performance of the five algorithms in estimated position error and energy consumption, it can be proved that the ASIA method of the proposed algorithm can greatly reduce energy consumption while providing good estimation performance.

Compared to the FLAS(FT) algorithm, the FLAS(DT) algorithm can dynamically adjust the uncertainty threshold according to the estimated position of the target. As shown in Fig. 16, as the target gradually approaches the left edge of the tracking area, the estimated uncertainty threshold is getting lower and lower. Different uncertainty thresholds can cause different estimated uncertainty and energy consumption. The estimated uncertainty of the FLAS(DT) algorithm also decreases with the dynamic threshold line. With the sampling moment 60s as the demarcation point, the left part of the FLAS(DT) algorithm has high uncertainty, few sampling steps, less energy consumption, while the right part has low uncertainty, more sampling steps, and more energy consumption. This can prove that the DUTA method can dynamically adjust the uncertainty threshold to alleviate the imbalance of energy consumption.

V. CONCLUSION AND FUTURE WORK

In this paper, an adaptive sampling algorithm for target tracking in UWSNs is proposed to maximize energy efficiency and balance energy consumption simultaneously. For maximizing energy efficiency, the adaptive sampling interval adjustment (ASIA) method uses a two-input- single-output fuzzy logic controller to adaptively adjust the sampling interval. For balancing energy consumption, the dynamic uncertainty threshold adjustment (DUTA) method uses a single-input-single-output fuzzy logic controller to dynamically adjust the uncertainty threshold in the ASIA method. The simulation results demonstrate that, compared to the existing adaptive sampling algorithm, the proposed algorithm not only improves the energy efficiency but also alleviates the imbalance of energy consumption.

In the future, we aim to study dynamic clustering methods and cluster node selection methods, in order to reduce the number of cluster nodes and energy consumption. This paper uses EKF as target position estimation method. EKF can only deal with zero mean gaussian noises, and more advanced methods are required for non-gaussian noises.

REFERENCES

[1] H.-P. Tan, R. Diamant, W. K. G. Seah, and M. Waldmeyer, "A survey of techniques and challenges in underwater localization," *Ocean Eng.*, vol. 38, nos. 14–15, pp. 1663–1676, Oct. 2011.

[2] A. Darehshoorzadeh and A. Boukerche, "Underwater sensor networks: A new challenge for opportunistic routing protocols," *IEEE Commun. Mag.*, vol. 53, no. 11, pp. 98–107, Nov. 2015.

[3] H. Bhambri and A. Swaroop, "Underwater sensor network: Architectures, challenges and applications," in *Proc. INDIACom*, New Delhi, India, Jun. 2014, pp. 915–920.

[4] C.-M. Chao, M.-W. Lu, and Y.-C. Lin, "Energy-efficient multichannel MAC protocol design for bursty data traffic in underwater sensor networks," *IEEE J. Ocean. Eng.*, vol. 40, no. 2, pp. 269–276, Apr. 2015.

[5] C. Kam, S. Kompella, G. D. Nguyen, A. Ephremides, and Z. Jiang, "Frequency selection and relay placement for energy efficiency in underwater acoustic networks," *IEEE J. Ocean. Eng.*, vol. 39, no. 2, pp. 331–342, Apr. 2014.

[6] A. A. Aziz, Y. A. Sekercioglu, P. Fitzpatrick, and M. Ivanovich, "A survey on distributed topology control techniques for extending the lifetime of battery powered wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 121–144, 1st Quart., 2012.

[7] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: Research challenges," *Ad Hoc Netw.*, vol. 3, no. 3, pp. 257–279, Mar. 2005.

[8] S. M. Dehnavi, M. Ayati, and M. R. Zakerzadeh, "Three dimensional target tracking via underwater acoustic wireless sensor network," in *Proc. Artif. Intell. Robot.*, Qazvin, Iran, Jun. 2017, pp. 153–157.

[9] R. W. L. Coutinho, A. Boukerche, L. F. M. Vieira, and A. A. F. Loureiro, "Geographic and opportunistic routing for underwater sensor networks," *IEEE Trans. Comput.*, vol. 65, no. 2, pp. 548–561, Feb. 2016.

[10] A. Kose and E. Masazade, "Adaptive sampling with sensor selection for target tracking in wireless sensor networks," in *Proc. 48th Asilomar Conf. Signals Syst. Comput.*, Pacific Grove, CA, USA, Nov. 2014, pp. 909–913.

[11] W. Xiao, J. K. Wu, and L. Xie, "Adaptive sensor scheduling for target tracking in wireless sensor network," *Proc. SPIE*, vol. 5910, pp. 59100B-1–59100B-9, Sep. 2005. [Online]. Available: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/5910/59100B/Adaptive-sensor-scheduling-for-target-tracking-in-wireless-sensor-network/10.1117/12.618124.short?SSO=1&tab=ArticleLink

[12] W. Xiao, L. Xie, J. Chen, and L. Shue, "Multi-step adaptive sensor scheduling for target tracking in wireless sensor networks," in *Proc. ICASSP*, Toulouse, France, Sep. 2006, pp. 705–708.

[13] W. Xiao, S. Zhang, J. Lin, and C. K. Tham, "Energy-efficient adaptive sensor scheduling for target tracking in wireless sensor networks," *J. Control Theory Appl.*, vol. 8, no. 1, pp. 86–92, 2010.

[14] J. Lin, F. Lewis, W. Xiao, and L. Xie, "Accuracy based adaptive sampling and multi-sensor scheduling for collaborative target tracking," in *Proc. ICARCV*, Singapore, Dec. 2006, pp. 1–6.

[15] J. Lin, W. Xiao, F. L. Lewis, and L. Xie, "Energy-efficient distributed adaptive multisensor scheduling for target tracking in wireless sensor networks," *IEEE Trans. Instrum. Meas.*, vol. 58, no. 6, pp. 1886–1896, Jun. 2009.

[16] G. Isbitiren and O. B. Akan, "Three-dimensional underwater target tracking with acoustic sensor networks," *IEEE Trans. Veh. Technol.*, vol. 60, no. 8, pp. 3897–3906, Oct. 2011.

[17] H. Luo, Z. Guo, K. Wu, F. Hong, and Y. Feng, "Energy balanced strategies for maximizing the lifetime of sparsely deployed underwater acoustic sensor networks," *Sensors*, vol. 9, no. 9, pp. 6626–6651, 2009.

[18] Q. Zhang, M. Liu, and S. Zhang, "Node topology effect on target tracking based on UWSNs using quantized measurements," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2323–2335, Oct. 2015.

[19] G. Han, C. Zhang, L. Shu, and J. J. P. C. Rodrigues, "Impacts of deployment strategies on localization performance in underwater acoustic sensor networks," *IEEE Trans. Ind. Electron.*, vol. 62, no. 3, pp. 1725–1733, Mar. 2015.

[20] D. Pompili, T. Melodia, and I. F. Akyildiz, "Three-dimensional and two-dimensional deployment analysis for underwater acoustic sensor networks," *Ad Hoc Netw.*, vol. 7, no. 4, pp. 778–790, 2009.

[21] H. Long, Z. Qu, X. Fan, and S. Liu, "Improved average consensus scalable algorithm of target tracking for wireless sensor network," in *Proc. 24th Chin. Control Decis. Conf. (CCDC)*, May 2012, pp. 3336–3341.

[22] M. Erol-Kantarci, H. T. Mouftah, and S. Oktug, "A survey of architectures and localization techniques for underwater acoustic sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 3, pp. 487–502, 3rd Quart., 2011.

[23] J. Yan, Z. Xu, Y. Wan, C. Chen, and X. Luo, "Consensus estimation-based target localization in underwater acoustic sensor networks," *Int. J. Robust Nonlinear*, vol. 27, no. 9, pp. 1607–1627, Jun. 2017.

[24] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation With Applications to Tracking and Navigation*. New York, NY, USA: Wiley, 2001, pp. 379–413.

[25] M. Poostpasand and R. Javidan, "An adaptive target tracking method for 3D underwater wireless sensor networks," *Wireless Netw.*, vol. 24, no. 8, pp. 2797–2810, Apr. 2017.

[26] M. Chu, H. Haussecker, and F. Zhao, "Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks," *Int. J. High Perform. Comput. Appl.*, vol. 16, no. 3, pp. 293–313, Aug. 2002.

[27] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. 33rd HICSS*, Wailea Maui, HI, USA, Jan. 2000, pp. 3005–3014.

[28] J. Yan, X. Yang, X. Luo, and C. Chen, "Energy-efficient data collection over AUV-assisted underwater acoustic sensor network," *IEEE Syst. J.*, to be published, doi: 10.1109/JSYST.2017.2789283.

[29] R. J. Urick, *Principles of Underwater Sound*. New York, NY, USA: McGraw-Hill, 1983.

[30] A. Sehgal, C. David, and J. Schönwälder, "Energy consumption analysis of underwater acoustic sensor networks," in *Proc. IEEE Conf. Oceans*, Waikoloa, HI, USA, Sep. 2011, pp. 1–6.

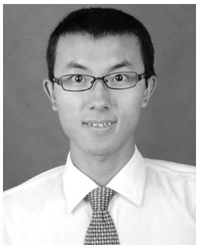
[31] J. M. J. Kartha and L. Jacob, "Lifetime enhancement in sparse underwater acoustic sensor networks using mobile elements," in *Proc. Int. Conf. Signal Process. Commun. (SPCOM)*, Jul. 2014, pp. 1–6.

[32] C. L. Lin, Y. M. Chang, C. C. Hung, C. D. Tu, and C. Y. Chuang, "Position estimation and smooth tracking with a fuzzy-logic-based adaptive strong tracking Kalman filter for capacitive touch panels," *IEEE Trans. Ind. Electron.*, vol. 62, no. 8, pp. 5097–5108, Aug. 2015.

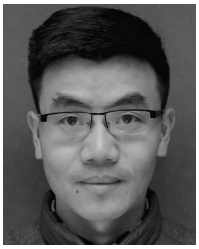
[33] J. Yan, X. Zhang, X. Luo, Y. Wang, C. Chen, and X. Guan, "Asynchronous localization with mobility prediction for underwater acoustic sensor networks," *IEEE Trans. Veh. Technol*, vol. 67, no. 3, pp. 2543–2556, Mar. 2018.

YANLONG SUN is currently pursuing the Ph.D. degree in control science and engineering with Yanshan Uinversity, Qinhuangdao, China. His current research interests include underwater wireless sensor networks and target tracking.

YAZHOU YUAN received the B.S. and M.S. degrees in control science and engineering from Yanshan University, Qinhuangdao, China, in 2009 and 2012, respectively, and the Ph.D. degree in control theory and engineering from Shanghai Jiaotong University in 2016. He is currently a Research Assistant with the Institute of Electrical Engineering, Yanshan University, China. His research interests include real-time location algorithm and industrial Internet of Things.



XIAOLEI LI received the B.Eng. degree in automation from Yanshan University, Qinhuangdao, China, in 2012, where he is currently pursuing the Ph.D. degree in control science and engineering. His current research interests include cooperative control of multi-agent systems and T-S fuzzy control system.



QIMIN XU is currently pursuing the Ph.D. degree with the Department of Automation, Shanghai Jiao Tong University. His research interests are in the control and optimization of wireless network systems, with a recent focus on the analytics and optimization of operations and economics of microgrids.

XINPING GUAN (M'02–SM'04–F'18) was a Professor and the Dean of electrical engineering at Yanshan University, China. He is currently a Chair Professor of Shanghai Jiao Tong University, China, where he is the Deputy Director of the University Research Management Office, and the Director of the Key Laboratory of Systems Control and Information Processing, Ministry of Education of the People's Republic of China.

His current research interests include industrial cyber-physical systems, wireless networking, and applications in smart city and smart factory, and underwater sensor networks. He has authored and/or co-authored four research monographs, over 270 papers in the IEEE Transactions and other peer-reviewed journals, and numerous conference papers. As the Principal Investigator, he has finished/been working on many national key projects. He is the Leader of the prestigious Innovative Research Team of the National Natural Science Foundation of China (NSFC). He is an Executive Committee Member of the Chinese Automation Association Council and the Chinese Artificial Intelligence Association Council. He received the First Prize of the Natural Science Award from the Ministry of Education of China in 2006 and 2016, respectively, and the Second Prize of the National Natural Science Award of China in 2008. He was a recipient of the IEEE Transactions on Fuzzy Systems Outstanding Paper Award in 2008. He is a National Outstanding Youth honored by NSF of China, the Changjiang Scholar by the Ministry of Education of China, and the State-level Scholar of New Century Bai Qianwan Talent Program of China.