# Multi-AUV Assisted Seamless Underwater Target Tracking Relying on Deep Learning and Reinforcement Learning

Anonymous Authors

***Abstract*—Since seamless tracking of the underwater target is crucial for various underwater applications, we propose a fusion algorithm combining deep learning and reinforcement learning for multi-autonomous underwater vehicles (AUVs) to seamlessly track the underwater target. The framework of our proposed fusion algorithm consists of two stages. In the first stage, we propose an underwater target localization method based on convolutional neural network (CNN) that relies on shaft-rate electric fields, in which the data collected by underwater sensors is utilized to train CNN to achieve accurate target localization. In the second stage, we innovatively propose a multi-agent soft actor-critic (MASAC) reinforcement learning algorithm based on centralized training with decentralized execution, in which appropriate reward functions are designed to encourage multiple AUVs to cooperate in seamlessly tracking the target in unknown environments while avoiding obstacles. Simulation results show that the proposed fusion algorithm has excellent performance, while the real-time target localization accuracy is 97.8%, and AUVs can carry out seamlessly cooperative tracking of the target in unknown environment.**

## I. Introduction

Using autonomous underwater vehicles (AUVs) to seamlessly track the underwater target is the key to enabling underwater applications such as underwater rescue and combat [1]. However, due to the high mobility of the target and complex underwater environment, the AUV can not accurately obtain the location of the target in real time, and often needs to rely on various localization methods. Moreover, the limited sensing range of a single AUV can not cope with the problem of target escape, so it's necessary to use multiple AUVs to track the target cooperatively, that is, multi-AUV forms an intelligent group through the underwater acoustic link and communications, improving the performance of target tracking. Therefore, how to achieve accurate target localization and stable tracking simultaneously has become an open challenge for seamless underwater target tracking [2].

Traditional target localization methods mainly rely on active or passive sonar systems [3], [4]. Nevertheless, due to the limited propagation ability of the sound signal, these techniques are insufficient in terms of localization range and precision, which are inadequate for the seamless tracking of the target [5]. In contrast, the target localization method based on the shaft-rate electric field can quickly and efficiently locate the underwater target [6]. The primary methods for shaft-rate electric field localization include precomputed direct current module-based localization, differential amplitude localization, Kalman filter localization, and odorless particle filtering localization [7]–[9]. However, these approaches have limitations, such as low accuracy and restricted applicability. Recently, deep learning methods have gained attention due to their potential. Li *et al.* [10] adopted deep learning technology to achieve automatic landing localization of drones, and Cebollada *et al.* [11] achieved robot localization through a combination of convolutional neural networks and computer vision. Inspired by the application of deep learning techniques to localization problems, we employ deep learning methods to study underwater target localization for the first time.

On the other hand, underwater target tracking is very challenging due to the lack of prior environmental information, the existence of obstacles, and the unknown motion state of the target [12]. Traditional target tracking methods for AUVs have been mainly relying on global information or by employing model-based or centralized control algorithms [13]. However, these methods are limited by the inability to know the future maneuvering information and behavioral strategies of the target, which leads to unexpected performance and adaptability. Fortunately, reinforcement learning (RL) has emerged and been applied in AUV target tracking, which demonstrates the ability to address complex tasks and anticipate future outcomes without extensive prior knowledge [14]. Furthermore, the advanced approach, multi-agent reinforcement learning (MARL), has proven effective at enhancing the scheduling and strategic planning of multi-AUV via joint learning and concerted action, which has demonstrated superior outcomes across a range of intricate applications [15].

Based on the above analysis, this work innovatively proposes a fusion algorithm based on deep learning and reinforcement learning for multi-AUV assisted seamless underwater target tracking. Specifically, we propose an shaft-rate electric field assisted localization method based on convolutional neural network (CNN) to accurately locate the target in a large range in real time. Moreover, we train multi-AUV using proposed multi-agent soft actor-critic (MASAC) to track the target collaboratively and seamlessly. Our fusion algorithm connects the two sub-tasks together and provides a basic paradigm for seamless underwater target tracking.

The paper is structured as follows. In Section II, we provide a detailed description of the system model, including the scenario and basic principles. In section III, we formulate the problem, present the constrained optimization objective in detail, while section IV covers the details of the fusion algorithm. Section V provides simulation results for evaluating

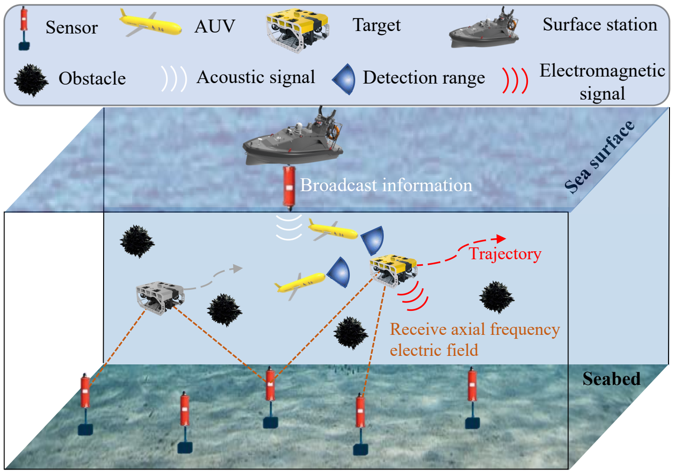

Fig. 1. Multi-AUV assisted seamless underwater target tracking scenario.

the rationality and performance of proposed fusion algorithm, followed by conclusions in Section VI.

## II. System Model

In this section, we describe the task scenario, assumptions, and model definitions used in this article. Fig. 1 depicts the task scenario, where target $T$ is randomly placed on a two-dimensional plane at depth of $d$, and the $N$ AUVs scattered around the starting point $O = (O_x, O_y, d)$ are ordered to cooperatively track the target. Coordinates of the target and AUV $i$ are defined as $p_T = (x_T, y_T, d)$ and $P_i = (x_i, y_i, d)$, respectively. The motion state of the target is unknown, and its location information can be obtained by the undewater sensor network using the shaft-rate electric field method, that is, the underwater sensor nodes receive the shaft-rate electric field generated by the target and transmit it to the surface base station through the acoustic signal, and then the location information of the target is solved and broadcast to the AUVs. Based on the target's location information broadcast by the base station, AUVs are ordered to coordinate target tracking.

### A. Dynamics of AUVs

Considering the two-dimensional research plane, we adopt a three-degree-of-freedom underdriven AUV model [15], for AUV $i$, it has a body-fixed coordinate system $\boldsymbol{v}_i = [v_{i,x}, v_{i,y}, \omega_i]^T$ and an earth-fixed reference system $\boldsymbol{\eta}_i = [x_i, y_i, \theta_i]^T$, where $v_{i,x}$, $v_{i,y}$ and $\omega_i$ represents the surge, sway and yaw velocities, respectively. In addition, $\theta_i$ is the yaw angle. Then, the dynamics of AUV $i$ is

$$\dot{\boldsymbol{\eta}}_i = \boldsymbol{J}(\boldsymbol{\eta}_i) + \boldsymbol{v}_i, \tag{1}$$

$$\boldsymbol{M}\dot{\boldsymbol{v}}_i + \boldsymbol{C}(\boldsymbol{v}_i)\boldsymbol{v}_i + \boldsymbol{D}(\boldsymbol{v}_i)\boldsymbol{v}_i + \boldsymbol{G}(\boldsymbol{\eta}_i) + \boldsymbol{\sigma}_i = \boldsymbol{\tau}_i, \tag{2}$$

where $\boldsymbol{M}$ denotes the mass matrix, $\boldsymbol{C}(\boldsymbol{v}_i)$ is the skew symmetrical matrix, denoting the centrifugal and Coriolis forces, $\boldsymbol{D}(\boldsymbol{v}_i)$ is the damping matrix describing viscous hydrodynamic force, while $\boldsymbol{G}(\boldsymbol{\eta}_i)$ represents the restoring forces of gravity and buoyancy. Moreover, $\boldsymbol{\tau}_i$ is the input control force, while $\boldsymbol{\sigma}_i$ denotes the forces induced by disturbances. And $\boldsymbol{J}(\boldsymbol{\eta}_i)$ denotes the transformation matrix, which is defined as

$$\boldsymbol{J}(\boldsymbol{\eta}_i) = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 \\ \sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

Considering the practical application, the above kinematics and dynamics equations need to be discretized [16], which can be expressed as

$$\boldsymbol{\eta}_{i,t+1} = \boldsymbol{\eta}_{i,t} + \Delta T \cdot \boldsymbol{J}(\theta_{i,t}) \cdot \boldsymbol{v}_{i,t}, \tag{4}$$

$$\boldsymbol{v}_{i,t+1} = \boldsymbol{v}_{i,t} + \Delta T \cdot \boldsymbol{M}^{-1}(\boldsymbol{\tau}_{i,t} - \boldsymbol{D}(\boldsymbol{v}_{i,t})\boldsymbol{v}_{i,t} - \boldsymbol{G}(\boldsymbol{\eta}_{i,t}) - \boldsymbol{\sigma}_{i,t}), \tag{5}$$

where subscript $t$ denotes the sampled values at time step $t \cdot \Delta T$. Hence, given the input $\boldsymbol{\tau}_t$ and the current location $\boldsymbol{\eta}_t$ and velocity $\boldsymbol{v}_t$, AUV's next location and velocity can be solved.

### B. AUV Communications and Target Detection

The detection of the target and communication between the AUVs are both accomplished through sonar. These processes can be consistently modeled using the underwater environment's active sonar equation [16]

$$EM = SL - 2TL(f, d) + TS - NL + DI - DT. \tag{6}$$

The unit of all parameters in Eq. (6) is dB, where $SL$, $TL$, $TS$, $NL$ and $DI$ represent the emission sound strength, transmission loss, target strength, environmental noise level and directionality index, respectively. $DT$ and $EM$ represent active sonar's detection threshold and echo margin, respectively. Furthermore, $TL$ is related to the detection radius $d$ and the center acoustic frequency $f$, i.e.

$$TL = 20\log(d) + da(f) \times 10^{-3}, \tag{7}$$

where $a(f)$ is the absorption coefficient, which can be expressed as

$$a(f) = 0.11\frac{f^2}{1+f^2} + 44\frac{f^2}{4100+f^2} + 2.75\times10^{-4}f^2 + 0.003. \tag{8}$$

Environmental noise $NL$ is composed of turbulence noise $Nt$, shipping noise $Ns$, wind noise $Nw$ and thermal noise $Nth$. The above noises can be expressed as Gaussian statistics, and the total power spectral density (PSD) of the $NL$ is

$$NL(f) = Nt(f) + Ns(f) + Nw(f) + Nth(f). \tag{9}$$

The noise components in Eq. (9) can be respectively described as

$$\begin{cases} 10\log N_t(f) = 17 - 30\log f, \\ 10\log N_s(f) = 30 + 20s + \log\left(f^{26}/(f+0.03)^{60}\right), \\ 10\log N_w(f) = 50 + 7.5\omega^{1/2} + 20\log\left(f/(f+0.4)^2\right), \\ 10\log N_{th}(f) = -15 + 20\log f, \end{cases} \tag{10}$$

where $s$ represents the shipping activity factor, and $w$ denotes the wind speed in meters per second (m/s), with $s$ ranging between 0 and 1.

### C. AUV Network Representation Relying on Graph Theory

According to graph theory, AUV network can be modeled as an undirected time-varying graph $G_{\boldsymbol{U}}(t)$, $\boldsymbol{U}$ is the set of AUVs. The dynamic change of communication link state between nodes can be represented by the Laplacian matrix $L_{\boldsymbol{U}}(t)$ of $G_{\boldsymbol{U}}(t)$. The calculation of the $L_{\boldsymbol{U}}(t)$ is given in Eq. (11), where $A_{\boldsymbol{U}}(t)$ and $D_{\boldsymbol{U}}(t)$ are the adjacency matrix and vertex degree diagonal matrix of the graph, respectively.

$$L_{Nu}(t) = \underbrace{\begin{bmatrix} d_1(t) & 0 & \cdots & n \\ 0 & d_2(t) & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & d_M(t) \end{bmatrix}}_{D_{\boldsymbol{U}(t)}} - \underbrace{\begin{bmatrix} a_{11}(t) & \cdots & a_{1M}(t) \\ a_{21}(t) & \cdots & a_{2M}(t) \\ \vdots & & \vdots \\ a_{M1}(t) & \cdots & a_{MM}(t) \end{bmatrix}}_{A_{\boldsymbol{U}(t)}}, \tag{11}$$

where $d_i(t) = \sum_{j=1}^{N}$, and $a_{ij}(t)$ represents the signal-to-noise ratio between AUV $i$ and AUV $j$ in underwater communications [16]

$$a_{ij}(t) = SL - TL - NL + DI. \tag{12}$$

By calculation, the element in the $i$-th row and $j$-th column of $L_{\boldsymbol{U}}(t)$ is defined as follows

$$L_{\boldsymbol{U}}(t)_{ij} = \begin{cases} -a_{ij}(t), & i \neq j, a_{ij}(t) \geq DT \\ \sum_{k=1,k\neq i}^{k=N} a_{ij}(t), & i = j, a_{ij}(t) \geq DT \\ 0. & i \neq j, a_{ij}(t) < DT \end{cases} \tag{13}$$

The second smallest eigenvalue of $L_{\boldsymbol{U}}(t)$, denoted as $\lambda_2(L_{\boldsymbol{U}}(t))$, is the algebraic connectivity of the graph, which can describe the connectivity of the graph. The larger the $\lambda_2(L_{\boldsymbol{U}}(t))$, the stronger the connectivity, and when $\lambda_2(L_{\boldsymbol{U}}(t)) < 0$, it means that the graph is no longer connected.

### D. Forward and Inversion of Shaft-Rate Electric Field

We acquire the training dataset by applying the forward formula to capture shaft-rate electric field at distinct positional points. Utilizing this dataset, we train the CNN receiving shaft-rate electric field data as input and generating output representing the location associated with the shaft-rate electric field, which enables the inverse of target location. The shaft-rate electric field generated by horizontal and vertical electric dipole sources at points $(x_s, y_s, z_s)$ in stratified conducting media are expressed as [17]

$$\begin{aligned} F'(x_s, y_s, z_s, x, y, z) &= P(x_s, y_s, x, y) \\ &\int_0^{\infty} \sum_{v=0}^{1} f_v(\sigma, z_s, z, \lambda) J_v(\lambda r) d\lambda, \end{aligned} \tag{14}$$

where $F'(x_s, y_s, z_s, x, y, z)$ represents the shaft-rate electric field generated by a unit horizontal electric dipole source. $P(x_s, y_s, x, y)$ is related to the horizontal position of the field source, the measuring point, and the dipole moment. $J_v(\lambda r)$ is a Bessel function of order $v$ equal to 0 or 1, $f_v(\sigma, z_s, z, \lambda)$ represents the corresponding kernel function, which depends on the conductivity $\sigma$, thickness of the conductive layer, and the vertical position of the field source and the measurement point. $r = \sqrt{(x - x_s)^2 + (y - y_s)^2}$ is the horizontal distance between the field source and the measuring point.

Usually, the shaft-rate electric field data is susceptible to contamination arising from external interference, and such contamination can be characterized by Gaussian random noise. The objective function for traditional gradient-based optimization under the $L_2$-norm condition is set as follows

$$\phi = \left\| \boldsymbol{W}_d(\boldsymbol{d} - \boldsymbol{F}(\boldsymbol{m})) \right\|^2, \tag{15}$$

where $\boldsymbol{F}(\boldsymbol{m})$ represents the forward operator, and $\boldsymbol{m}$ is the model reference vector. $\boldsymbol{d}$ is the inversion data vector, consisting of the real and imaginary parts of the electric field components, which are disturbed by the Gaussian random noise conforming to the normal distribution. In this study, Gaussian random noise is set as follows

$$G = A \times \eta \times r, \tag{16}$$

where $A$ is the strength of the electric field, $\eta$ represents the added noise intensity, and $r \in [-1, 1]$ is a random number.

In addition, $\boldsymbol{W}_d$ is the data weighting matrix, which can be expressed as follows [18]

$$\boldsymbol{W}_d = \text{diag}\left(\frac{1}{|d_i| r_i + \eta_m}\right);\ i = 1, \cdots, N_d, \tag{17}$$

where $r_i$ is the estimated relative error at ith datum plane. A small constant $\eta_m$ corresponding to the noise observation lower bound of the data is added to prevent the inversion from overemphasizing the low-amplitude data. And $N_d$ represents the number of observation data.

## III. Problem Formulation

In this section, we first model the multi-AUV assisted seamless underwater target tracking problem as a partially observable Markov game process (POMGP). Then, the constrained optimization objective is presented and the reward function is designed in detail.

### A. Partially Observable Markov Game Process Modeling

The process of seamless underwater target tracking can be modeled as a POMGP, which consists of the following tuple

$$\mathcal{M} = (\boldsymbol{\mathcal{S}}, \boldsymbol{O}, \boldsymbol{\mathcal{A}}, P, R(t), \gamma, \rho), \tag{18}$$

where $\boldsymbol{\mathcal{S}}$ stands for all possible state spaces of $N$ AUVs, which represents the global information, while $\boldsymbol{O}_i$ represents the observation space of AUV $i$. For AUV $i$, the action space is $\boldsymbol{\mathcal{A}}_i$. The state transition function of the environment is $P : \boldsymbol{\mathcal{S}} \times \boldsymbol{\mathcal{A}}_1 \times \cdots \times \boldsymbol{\mathcal{A}}_i \rightarrow \boldsymbol{\omega}(\boldsymbol{\mathcal{S}})$, while the reward function of the AUV is $R(t) : \boldsymbol{\mathcal{S}} \times \boldsymbol{\mathcal{A}} \rightarrow R(t)$, whose specific design is given in the following subsection, and $\gamma \rightarrow [0, 1]$ is the discount factor, $\rho : \boldsymbol{\mathcal{S}} \rightarrow [0, 1]$ is initial state distribution. Besides, AUV $i$ owns the policy $\pi_i : \boldsymbol{\mathcal{O}}_i \times \boldsymbol{\mathcal{A}}_i \rightarrow [0, 1]$, which is a probability distribution representing the probability that the AUV will take each action for each observation. And the partially observed information of AUV $i$ from the global state is: $\boldsymbol{o}_i : \boldsymbol{\mathcal{S}} \rightarrow \boldsymbol{O}_i$.

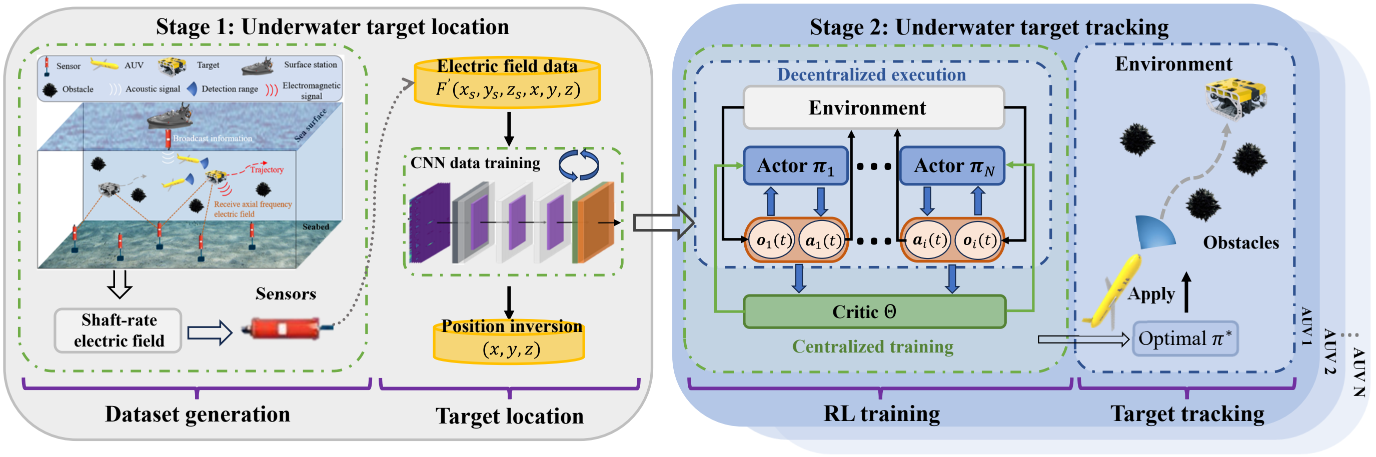

Fig. 2. The framework of our proposed fusion algorithm for seamless tracking of the underwater target, which includes two stages: underwater target localization and tracking. In the first stage, shaft-rate electric field data is collected by sensors to generate datasets, which are used to train the CNN to locate the target's position. In the second stage, we utilize the MASAC algorithm under centralized training decentralized execution (CTDE) framework to train multi-AUV to learn optimal policy, aiming to improve target tracking performance.

In addition, in the target localization and cooperative tracking problem, AUV $i$ will try to make the proper action $\boldsymbol{a}_i(t)$ according to its observation $\boldsymbol{o}_i(t)$ and policy $\pi_i(\boldsymbol{a}_i(t)\,|\boldsymbol{o}_i(t)\,)$ to maximize discount reward $J(\theta_i)$. Based on the above intuition, the constrained optimization objective can be formulated as

$$\max_{\pi_i} J(\theta_i) = \max_{\pi_i} E\left[\sum_{t=t'}^{T=\infty} \gamma^{t-t'} R(t)\right], \tag{19}$$

$$v_{\min} \leq \|\boldsymbol{v}_i(t)\| \leq v_{\max}, \omega_{\min} \leq \|\boldsymbol{\omega}_i(t)\| \leq \omega_{\max}, \forall i \in N, \tag{20}$$

where $\theta_i$ stands for the policy parameters of the AUV $i$, while $\|\boldsymbol{v}_i(t)\| \in [v_{\min}, v_{\max}]$ and $\|\boldsymbol{\omega}_i(t)\| \in [\omega_{\min}, \omega_{\max}]$ denotes the velocity and angular velocity, respectively.

### *B. Design of Reward Function*

The AUV adjusts its policy according to the reward obtained by the current action, so it is crucial to design an appropriate reward function. The appropriate reward function should be designed to guide the multi-AUV to complete the task. Taking the engineering practice into account, the reward function $R(t)$ can be divided into the following optimization objectives:

**Collision avoidance**: To ensure the safety of multi-AUV in the task, the distance between AUVs and between AUVs and obstacles should not be less than $l_{\min}^{i\leftrightarrow j}$. To avoid the collision, we need to set $l_{\min}^{i\leftrightarrow j}$ as the safe distance and design the reward function $r_A(t)$ to punish collision

$$r_A(t) = -500\,\text{ceil}\left(l_{\min}^{i\leftrightarrow j} / \min(\boldsymbol{l}(t))\right), \forall i, j \leq N, i \neq j, \tag{21}$$

where $\boldsymbol{l}(t) = \{l_1(t), \cdots, l_i(t)\}$ represents the distance vector, in which $l_i(t)$ is the distance from AUV $i$ to the nearest obstacle. And ceil is the integer up function.

**Keep tracking**: To ensure each AUV sustainably carryout tracking tasks throughout the whole process, we set $l_{\max}^{i\leftrightarrow T}$ as the target distance and reward AUV $i$'s tracking results

$$r_T(t) = 1000\,\text{ceil}(l_{\max}^{i\leftrightarrow T} / l^{i\leftrightarrow T}(t)), i = 1, \cdots, N, \tag{22}$$

where $l^{i\leftrightarrow T}$ is the distance between AUV $i$ and the target.

**Encourage exploration**: Based on the intuition that the AUVs need to be prevented from randomly wandering when exploring and interacting in the environment, we use $r_E$ to encourage AUV $i$ to move towards the target

$$r_E(t) = \begin{cases} 2, & l^{i\leftrightarrow T}(t-1) > l^{i\leftrightarrow T}(t), \\ -2, & l^{i\leftrightarrow T}(t-1) < l^{i\leftrightarrow T}(t), \end{cases} i = 1, \cdots, N\,. \tag{23}$$

**Maintain connectivity**: Based on the previous analysis, we can use reward $r_C(t)$ to encourage the multi-AUV to maintain connectivity. When $r_C(t)$ is greater than 0, the AUV network is connected, which can significantly improve the probability of target tracking, and the higher the value, the better the connectivity. When $r_C(t)$ is less than 0, the AUV network is no longer fully connected, and the system gets the corresponding punishment

$$r_C(t) = \lambda_2 L_{\boldsymbol{U}}(t). \tag{24}$$

In summary, weighing all the factors, the total reward function $R(t)$ can be expressed as

$$R(t) = \varepsilon_A r_A(t) + \varepsilon_T r_T(t) + \varepsilon_E r_E(t) + \varepsilon_C r_C(t), \tag{25}$$

where $\varepsilon_A$, $\varepsilon_T$, $\varepsilon_E$, and $\varepsilon_c$ are the weight coefficients corresponding to reward functions $r_A(t), r_T(t), r_E(t)$ and $r_C(t)$, respectively.

## IV. Target Localization and Tracking Algorithms

Seamless tracking of the underwater target includes two sub-tasks: target localization and target tracking. The CNN is firstly adopted for training based on the collected shaft-rate electric field data to achieve accurate target localization. Meanwhile, the MASAC algorithm is further adopted to train AUVs to learn optimal policies and improve the collaboration performance of tracking. The proposed algorithm' framework is depicted in Fig. 2.

### *A. Location Inversion Based on CNN*

The neural network model used in this study is the CNN, which can be divided into two parts: the convolutional layer and fully connected layer. In the convolutional layer, the maximum pooling layer is set to reduce the amount of data computation. In the fully connected layer, the input data is multiplied and added with the parameter $b_l$ to add the bias as the output of this layer, which can be expressed as follows

$$v_k = \sum_k w_{kl} h_k + h_l,\ k = 1, \cdots, n, l = 1, \cdots, m, \tag{26}$$

where $h_k$ is the hidden layer node ($0 < k \leq n$), $v_k$ is the visible layer node ($0 < k \leq m$), and $w_{kl}$ is the connection weight matrix between the visible layer and the hidden layer. The hidden layer output can be calculated as follows

$$z_l = f_a(\sum_k w_{kl} h_k + b_l). \tag{27}$$

Then, the softmax layer is set up after the fully connected layer to convert feature vectors into probability vectors for multiple data classifications, which is shown as follows

$$q_k = \frac{\exp(z_k)}{\sum_{l=1}^{\mathcal{N}} \exp(z_l)}, \tag{28}$$

where $z_k$ represents the input data to the softmax layer, and $q_k$ denotes the output data from the softmax layer, which satisfies $\Sigma_{\mathrm{k}=1}^{\mathcal{N}} q_k = 1$. Thus, the simulation of probability is realized. The softmax layer rear connection cross entropy function is

$$E = -\sum_{k=1}^{\mathcal{N}} p_k \times \log(q_k), \tag{29}$$

where $p_k$ is the label of the data.

Furthermore, to enforce network nonlinearity, and smoothness aiding in optimization, we employ the activation function DSoft, which is unbounded above, bounded below, non-monotonic, and smooth, avoiding saturation and gradient vanishing. The DSoft function can be expressed as follows

$$f(x) = x\xi(\ln(1 + e^x)), \tag{30}$$

where $\xi(x) = x(1+|x|)-1$ is the Softsign activation function, and the derivative of DSoft can be expressed as

$$f^{'}(x) = x^{-1} f(x) + x g(x), \tag{31}$$

$$g(x) = (1 + e^{-x})^{-1}(\ln(1 + e^x) + 1)^{-2}. \tag{32}$$

Finally, we utilize the Adam algorithm, an optimizer combining Momentum and RMSProp benefits. By sampling a minibatch of $p$ examples from the training set $\{x^{(1)}, \ldots, x^{(p)}\}$ with corresponding targets $y^{(n)}$, the gradient can be computed as

$$g \leftarrow \frac{1}{l} \nabla\theta \sum_n L(f(x^{(n)}; \theta), y^{(n)}). \tag{33}$$

Then, update biased first and second moment estimate

$$\lambda \leftarrow \xi_1 \lambda + (1 - \xi_1 g), \tag{34}$$

$$\varsigma \leftarrow \xi_2 \xi + (1 - \xi_2 g) \odot g, \tag{35}$$

where $\xi_1$ and $\xi_2$ are the exponential decay rates for monent estimates, while $\lambda$ and $\varsigma$ are the first and second moment variables, respectively, and $\theta$ is a parameter that needs to be initialized. Hence, it is possible to rectify bias in both the first and second moments through the computation of the update. The process of achieving the gradient update entails the repetition of the aforementioned cycle.

### *B. Multi-Agent Soft Actor-Critic*

To successfully complete the multi-AUV seamless target tracking task, we adopt the MASAC algorithm based on the CTDE framework to train AUVs to learn optimal tracking policies. In CTDE, all agents share a centralized critic network, and when it comes to each agent, it has local actor network to make decisions $\boldsymbol{a}_i$ based on individual observations $\boldsymbol{o}_i$ and then gain reward $R_t$. Specifically, the goal of RL can be expressed as a constrained optimization problem

$$\begin{cases} \max_{\boldsymbol{\pi}} J(\boldsymbol{\theta}) = \max_{\boldsymbol{\pi}} E\left[\sum_{t=1}^{T=\infty} R\left(\boldsymbol{x}, \boldsymbol{a}_1, \cdots, \boldsymbol{a}_N\right)\right], \\ E_{\boldsymbol{o}_i \sim \rho^{\pi_i}, \boldsymbol{a}_i \sim \pi_i}\left(-\log \pi_i\left(\boldsymbol{a}_i \mid \boldsymbol{o}_i\right)\right) \geq H_0, \end{cases} \tag{36}$$

where $\boldsymbol{\pi}$, $\boldsymbol{\theta}$ and $\boldsymbol{x}$ represent the policies, parameters of the actor network and observations of all the agents, and can be expressed as $\boldsymbol{\pi} = \{\pi_1, \cdots, \pi_N\}$, $\boldsymbol{\theta} = \{\theta_1, \cdots, \theta_N\}$, and $\boldsymbol{x} = (\boldsymbol{o}_1, \cdots, \boldsymbol{o}_N)$, respectively. $H_0$ is entropy to help agents flexibly balance exploration and exploitation. Specifically, the actor network for each agent can be updated by Eq. (37)

$$\nabla_{\theta_i} J = E_{\boldsymbol{o}_i \sim \rho^{\pi_i}, \boldsymbol{a}_i \sim \pi_i}\left[\nabla_{\theta_i} \log \pi_i\left(\boldsymbol{a}_i \mid \boldsymbol{o}_i\right) Q_{\Theta_i}^{\pi_i}(\boldsymbol{x}, \boldsymbol{a}_1, \cdots, \boldsymbol{a}_N)\right], \tag{37}$$

where $\rho^{\pi_i}$ represents the observation distribution in the case of policy $\pi_i$, and $Q_{\Theta_i}^{\pi_i}(\boldsymbol{x}, \boldsymbol{a}_1, \cdots, \boldsymbol{a}_N)$ denotes the action value function.

In addition, to address the issue of $Q$ value overestimation, MASAC utilizes two critic networks, $\Theta_{1_i}$ and $\Theta_{2_i}$, as well as their respective target networks, $\Theta_{1_i}^{-}$and $\Theta_{2_i}^{-}$. Consequently, the loss function of $Q$ can be formulated as

$$\begin{aligned} L\left(\Theta_{1_i}\right) = E_{(\boldsymbol{x}, \boldsymbol{a}_1, \cdots, \boldsymbol{a}_N, R_t, \boldsymbol{x}') \sim \mathcal{R}}[(Q_{\Theta_{1_i}}^{\pi_i}\left(\boldsymbol{x}, \boldsymbol{a}_1, \cdots, \boldsymbol{a}_N\right) - \\ (R_t + \gamma V_{\Theta_{1_i}^{-}}(\boldsymbol{x}')))^2], \end{aligned} \tag{38}$$

$$\begin{aligned} L\left(\Theta_{2_i}\right) = E_{(\boldsymbol{x}, \boldsymbol{a}_1, \cdots, \boldsymbol{a}_N, R_t, \boldsymbol{x}') \sim \mathcal{R}}[(Q_{\Theta_{2_i}}^{\pi_i}\left(\boldsymbol{x}, \boldsymbol{a}_1, \cdots, \boldsymbol{a}_N\right) - \\ (R_t + \gamma V_{\Theta_{2_i}^{-}}(\boldsymbol{x}')))^2], \end{aligned} \tag{39}$$

where $\mathcal{R}$ represents the replay buffer to store collected data, while $V_{\Theta_{1_i}^{-}}$ and $V_{\Theta_{2_i}^{-}}$ represent the state value function with parameters $\Theta_{1_i}^{-}$ and $\Theta_{2_i}^{-}$, respectively. To prevent the AUV $i$ from getting stuck in local optimal policy, we introduce entropy regularization and express $V_{\Theta_{1_i}^{-}}(\boldsymbol{x}')$ and $V_{\Theta_{2_i}^{-}}(\boldsymbol{x}')$ as follows

$$\begin{aligned} V_{\Theta_{1_i}^{-}}(\boldsymbol{x}') = \min_{j=1,2} Q_{\Theta_{j_i}^{-}}^{\pi_i}\left(\boldsymbol{x}', \boldsymbol{a}_1', \cdots, \boldsymbol{a}_N'\right) - \\ \alpha_i \log \pi_i\left(\boldsymbol{a}_1', \cdots, \boldsymbol{a}_N' \mid \boldsymbol{x}'\right), \end{aligned} \tag{40}$$

**Algorithm 1:** Fusion Algorithm

```
Initialize the replay buffer $\mathcal{R}$, centralized critic network, target critic network, and actor network parameters $\Theta_{1_i}$, $\Theta_{2_i}$, $\Theta_{1_i}^-$, $\Theta_{2_i}^-$, $\theta_i$ of AUV $i$.
for each epoch $k$ do
    Reset the training environment and parameters.
    for each time step $t$ do
        for each AUV $i$ do
            Locate the target using trained model of CNN and shaft-rate electric field data.
            Sample an action according to the policy: $\boldsymbol{a}_i \sim \pi_i(\boldsymbol{a}_i|\boldsymbol{o}_i)$
            Obtain reward $R_t$ and the next state $\boldsymbol{x}'$ from environment.
            Store tuple $(\boldsymbol{x}, \boldsymbol{a}_1, \cdots, \boldsymbol{a}_N, R_t, \boldsymbol{x}')$ into $\mathcal{R}$
        end
        for each AUV $i$ do
            Extract $N$ tuples of data $(\boldsymbol{x}_n, \boldsymbol{a}_{1_n}, \cdots, \boldsymbol{a}_{N_n}, R_{t_n}, \boldsymbol{x}'_n)_{n=1,\cdots,N}$ from $\mathcal{R}$.
            Calculate $y_n = R_{t_n} + \gamma V_{\Theta_i}(\boldsymbol{x}'_n)$
            Update critic network by minimizing the loss function:
                $L(\Theta_i) = \frac{1}{N}\sum_{n=1}^{N}\left(Q_{\Theta_i}^{\pi_i}(\boldsymbol{x}_n, \boldsymbol{a}_{1_n}, \cdots, \boldsymbol{a}_{N_n}) - y_n\right)^2$
            Update actor network using the policy gradient:
                $\nabla_{\theta_i} J = \frac{1}{N}\sum_{n=1}^{N}\left[\nabla_{\theta_i} \log \pi_i(\boldsymbol{a}_{i_n} \mid \boldsymbol{o}_{i_n}) Q_{\Theta_i}^{\pi_i}(\boldsymbol{x}_n, \boldsymbol{a}_{1_n}, \cdots, \boldsymbol{a}_{N_n})\right]$
        end
        Soft update target network parameters.
    end
end
```

$$V_{\Theta_{2_i}^-}(\boldsymbol{x}') = \min_{j=1,2} Q_{\Theta_{j_i}^-}^{\pi_i}(\boldsymbol{x}', \boldsymbol{a}'_1, \cdots, \boldsymbol{a}'_N) - \alpha_i \log \pi_i(\boldsymbol{a}'_1, \cdots, \boldsymbol{a}'_N \mid \boldsymbol{x}'), \tag{41}$$

where $\alpha_i$ stands for the regularization coefficient, determining the weight placed on entropy in the policy. Subsequently, the policy's loss function can be derived from the simplified KL divergence

$$L_{\pi_i}(\theta_i) = E_{\boldsymbol{o}_i \sim \rho^{\pi_i}, \boldsymbol{a}_i \sim \pi_i}[\alpha_i \log(\pi_i(\boldsymbol{a}_1, \cdots, \boldsymbol{a}_N \mid \boldsymbol{o}_i)) - \min_{j=1,2} Q_{\Theta_i}^{\pi_i}(\boldsymbol{x}, \boldsymbol{a}_1, \cdots, \boldsymbol{a}_N)]. \tag{42}$$

## V. Simulation Results

In this section, we will verify the outstanding performance of our proposed fusion algorithm by simulating the whole process of training, which can be divided into two stages: target localization and tracking. Finally, the results of simulation experiments will be described and analyzed.

TABLE I
Parameters of Simulation Experiments

| **Parameters** | **Values** |
|---|---|
| AUV number $N$ | 2 |
| Max velocity $v_{\max}$ | 3.0 m/s |
| Max acceleration $a_{\max}$ | 5.0 m/s$^2$ |
| Max angular velocity $\omega_{\max}$ | 1.6 rad/s |
| Experimental site size 1 | 4000m × 4000m |
| Experimental site size 2 | 50m × 50m |
| Initial positions of AUVs | (-18.0, 20.5), (-18.0, 17.0) |
| Initial positions of the target | (-14.5, 20.5) |
| Locations of obstacles | (-20,10), (-10,-15), (0,10), (15,-15) |
| Safe distance $l_{\min}^{i \leftrightarrow j}$ | 1.5 m |
| Target distance $l_{\max}^{i \leftrightarrow T}$ | 2.5 m |
| Convolution kernel size | 4×4(1∼2 layers)/8×8(3∼5 layers) |
| Stabilization constant $\sigma$ | $10^{-8}$ |
| Learning rate $\lambda$ | $3 \times 10^{-4}$ |
| Reward weight coefficients | $\varepsilon_A$=1, $\varepsilon_T$=1,$\varepsilon_E$=1,$\varepsilon_C$=0.01 |
| Discount factor $\gamma$ | 0.99 |
| Soft updating rate $\eta$ | 0.01 |
| Regularization coefficient $\alpha$ | 0.2 |
| Training epochs $k$ | 100 |
| Replay buffer size $\mathcal{R}$ | $5 \times 10^4$ |
| Sample batch size | 256 |
| Hidden layer size | 256 |

### A. Simulation Settings

In the simulation experiment, the parameters can be divided into four parts: model parameters, simulation parameters, localization and track algorithm parameters. The parameters mentioned in the above are summarily shown in Table I.

### B. Experiment Process

The modeling area for simulation experiments is within the range of 4000m×4000m in the plane, and the seafloor sensors are placed evenly along x-axis to receive the shaft-rate electric field signals generated by the underwater targets. Then, 10% Gaussian random noise is added to the forward-modeling shaft-rate electric field data as the datasets for target localization. Finally, the data generated is utilized for the CNN model training, while contrast experiments are also conducted via commonly used Elman and back propagation (BP) models simultaneously.

As can be seen from the training results shown in Fig. 3, the accuracy and loss curves of training vary with the number of epochs, and finally tend to converge, respectively. Compared to Elman, the training process of the CNN model is apparently more stable. And the final localization accuracy is up to 97.8%, which achieves more superior performance than that of both Elman and BP.

In addition, we also conduct ablation experiments by changing the data batch size during training and the noise level in the shaft-rate electric field dataset to compare the performance and robustness of the algorithm under different conditions. As depicted in Fig. 4, under the same epoch, with the increase of batch size, the efficiency of CNN training continuously improves, and the training process is increasingly stable. Besides, upon observing Fig. 5, it can be discerned that as the noise level within the dataset escalates, there is a marginal decline in training efficiency of the CNN model. However, this

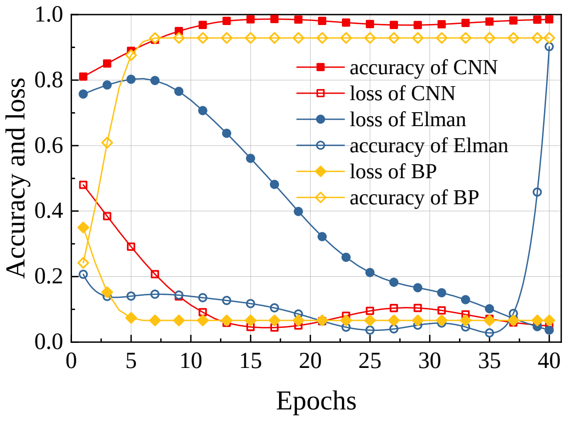

Fig. 3. The accuracy and loss curves of the CNN, Elman and BP models.

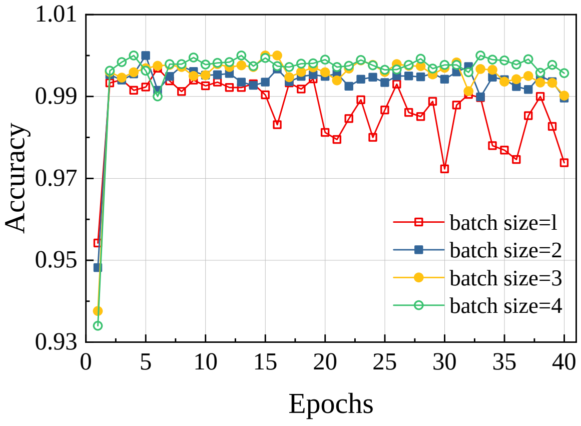

Fig. 4. The accuracy curves of target localization relying on CNN for training, with the batch size ranging from 1 to 4.

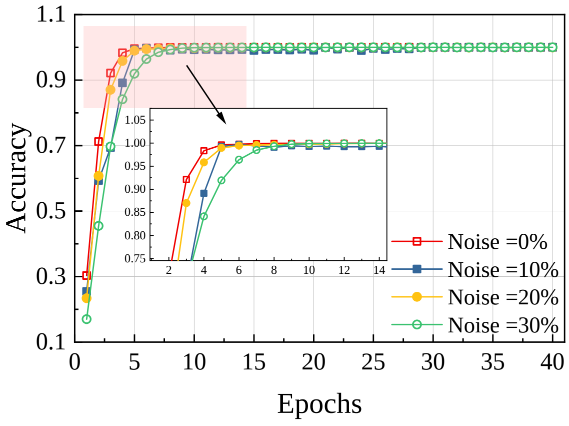

Fig. 5. The accuracy curves of target localization relying on CNN for training, with the noise level ranging from 0% to 30%.

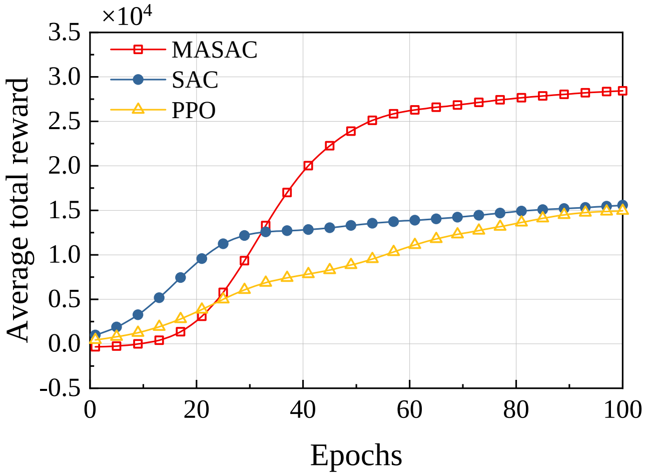

Fig. 6. Average total reward curves of MASAC, SAC, and PPO algorithms, while utilizing MASAC to train two AUVs, and SAC with PPO to train one single AUV for target tracking, respectively.

decline does not significantly impair the overall performance, as the accuracy of localization remains considerably high, which further demonstrates the outstanding robustness of the algorithm.

Then in the second stage of simulation, the size of the experiment site is 50m×50m and the target's initial position is uniformly set. The experiment includes two main stages: localization and tracking. In the localization stage, our goal is to realize the localization of the target. Preprocessed shaft-rate electric field data and target position are used for training and evaluating the performance of the CNN model. With the increase of the training epochs, accuracy rate and loss curves rise and decrease, and eventually tend to converge, respectively.

While the CNN model is well-trained, it will be deployed on each AUV for the second stage of the experiment. Training is first carried out by localizing the position of the target. Assuming that AUVs communicate their positions to each other in real-time via the base station on the surface, the MASAC algorithm is used to train two AUVs, tracker 1 and tracker 2, in the underwater scenario with obstacles to learn an optimal policy for tracking a moving target cooperatively. When the AUVs approach the target closely, its position is randomly reset, and the AUVs receive the reward. Due to the lack of prior experience and trial-and-error process, the reward curve usually fluctuates as the AUVs interact with the environment. With the increase of training epochs, the volatility of the reward curve decreases, indicating that AUVs have learned a relatively better policy and achieved a stable return from the environment. In addition, we also conduct contrast experiments through commonly used soft actor-critic (SAC) and proximal policy optimization (PPO) algorithms to train one single AUV for investigating the effect of AUV number and different algorithms on the performance of target tracking. As shown in Fig. 6, after training for 100 epochs, the average total reward of two AUVs via MASAC is almost double that of one single AUV relying on SAC or PPO, which highlights the superiority of collaboration between multi-AUV. Furthermore, through analyzing the outcomes of the three

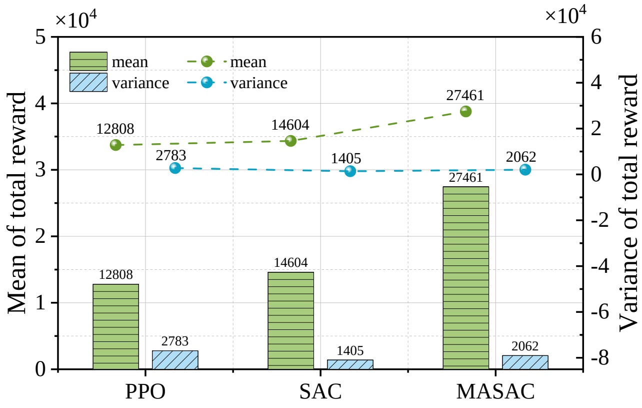

Fig. 7. The mean and variance of total reward data for the last 50 epochs obtained by utilizing PPO, SAC and MASAC for reinforcement learning training, respectively.

different algorithms, it has been obvious that in order to attain a state of convergence, PPO necessitates an extended number of training epochs when contrasted with MASAC and SAC, which highlights the superior sample and training efficiency inherent to MASAC and SAC. In addition, we also compare the mean and variance of total reward data for the last 50 epochs obtained by utilizing the three different algorithms in Fig. 7, which further highlights the higher mean reward and relatively lower variance of MASAC. Consequently, the findings indicate that employing MASAC for the training of AUVs leads to enhanced performance in target tracking tasks.

Finally, combined with the trained CNN model, the optimal policies are applied for the multi-AUV to realize the target localization and tracking tasks. The initial positions of AUVs and target are $(-18.0,\ 20.5)$, $(-18.0,\ 17.0)$, $(-14.5,\ 20.5)$, respectively. And the corresponding final positions are respectively distributed at $(-18.0,\ -17.0)$, $(-17.0,\ -21.0)$ and $(19.5,\ -16.0)$. The trajectories of AUVs and the target are shown in Fig. 8. Based on the previous description, multi-AUV successfully completes the task, which demonstrates the superiority of the proposed fusion algorithm in solving the problem of seamless underwater target tracking.

## VI. Conclusion

In this paper, we present a fusion algorithm for multi-AUV assisted seamless underwater target tracking. Shaft-rate electric field data is firstly collected to train CNN to invert the target's location, and then we formulate the target tracking task as an POMGP, designing reward functions and using the MASAC algorithm based on the CTDE framework to train multi-AUV to learn optimal tracking policies. Simulation results and contrast experiments indicate the outstanding localization accuracy and target tracking performance. The localization accuracy achieves 97.8%, higher than that of Elman and BP, while the total reward obtained via MASAC for training is higher than that of PPO and SAC, which enhance multi-AUV to accomplish the seamless target tracking task and obstacles avoidance, demonstrating the superiority of the proposed fusion algorithm and collaboration among AUVs.

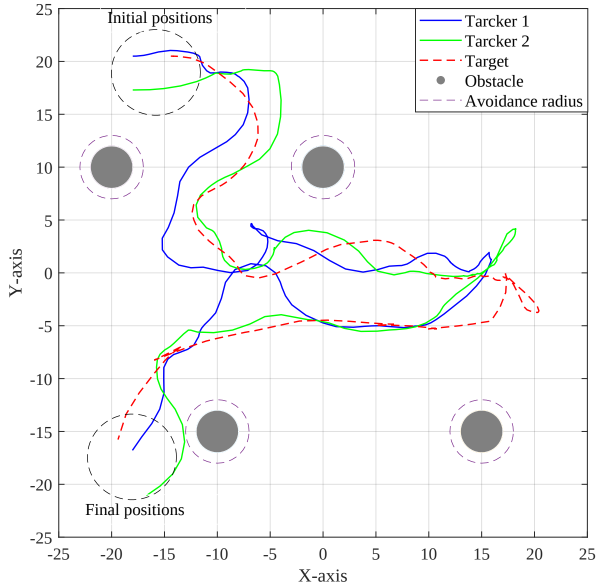

Fig. 8. The trajectories of AUVs and the target in the target localization and tracking task.

## References

[1] J. Wu, C. Song, J. Ma, J. Wu and G. Han, "Reinforcement Learning and Particle Swarm Optimization Supporting Real-Time Rescue Assignments for Multiple Autonomous Underwater Vehicles," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6807-6820, July. 2022.

[2] Y. Li, K. Cai, Y. Zhang, Z. Tang and T. Jiang, "Localization and Tracking for AUVs in Marine Information Networks: Research Directions, Recent Advances, and Challenges," in *IEEE Network*, vol. 33, no. 6, pp. 78-85, Nov.-Dec. 2019.

[3] R. Su, Z. Gong, C. Li and X. Shen, "Algorithm Design and Performance Analysis of Target Localization Using Mobile Underwater Acoustic Array Networks," in *IEEE Transactions on Vehicular Technology*, vol. 72, no. 2, pp. 2395-2406, Feb. 2023.

[4] X. Pan, Y. Shen, and J. Zhang, "IoUT based underwater target localization in the presence of time synchronization attacks," in *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3958–3973, Jun. 2021.

[5] X. Cao, D. Zhu, and S. X. Yang, "Multi-AUV target search based on bioinspired neurodynamics model in 3-D underwater environments," in *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 11, pp. 2364–2374, Nov. 2016.

[6] H. Wang, X. Wang and J. Zhang, "Research on positioning of Surface Warships in Shallow Water based on Circular Array," in *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, Chongqing, China, 2019, pp. 450-453.

[7] H. Lee, H. K. Jung, S. H. Cho, Y. Kim, H. Rim, and S. K. Lee, "Real-time localization for underwater moving object using precalculated DC electric field template," in *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 5813–5823, Oct. 2018.

[8] A. Noureldin, T. B. Karamat, M. D. Eberts and A. El-Shafie, "Performance Enhancement of MEMS-Based INS/GPS Integration for Low-Cost Navigation Applications," in *IEEE Transactions on Vehicular Technology*, vol. 58, no. 3, pp. 1077-1096, March. 2009.

[9] X. Chen, A. Marjovi, J. Huang and A. Martinoli, "Particle Source Localization With a Low-Cost Robotic Sensor System: Algorithmic Design and Performance Evaluation," in *IEEE Sensors Journal*, vol. 20, no. 21, pp. 13074-13085, Nov, 2020.

[10] M. Li and T. Hu, "Deep learning enabled localization for UAV autolanding," in *Chinese Journal of Aeronautics.*, vol. 34, no. 5, pp. 585-600, May. 2021.

[11] S. Cebollada, L. Payá, M. Flores, V. Román, A. Peidró and Ó. Reinoso, "A Deep Learning Tool to Solve Localization in Mobile Autonomous Robotics," in *2020 IEEE International Conference on Informatics in Control, Automation, and Robotics (ICINCO).* Portugal, 2021, pp. 232-241.

[12] X. Cao, L. Ren and C. Sun, "Dynamic Target Tracking Control of Autonomous Underwater Vehicle Based on Trajectory Prediction," in *IEEE Transactions on Cybernetics*, vol. 53, no. 3, pp. 1968-1981, Mar. 2023.

[13] V. Mnih, K. Kavukcuoglu, D. Silver, et al., "Human-level control through deep reinforcement learning," in *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[14] H. Wu, S. Song, K. You and C. Wu, "Depth Control of Model-Free AUVs via Reinforcement Learning," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 12, pp. 2499-2510, Dec. 2019.

[15] W. Wei, J. Wang, Z. Fang, J. Chen, Y. Ren, and Y. Dong, "3U: Joint design of UAV-USV-UUV networks for cooperative target hunting," in *IEEE Trans. Veh. Technol.*, vol. 72, no. 3, pp. 4085–4090, Mar. 2023.

[16] Z. Fang, J. Wang, J. Du, X. Hou, Y. Ren and Z. Han, "Stochastic Optimization-Aided Energy-Efficient Information Collection in Internet of Underwater Things Networks," in *IEEE Internet Things J.*, vol. 9, no. 3, pp. 1775-1789, Feb. 2022.

[17] Y. Li and G. Li, "Electromagnetic field expressions in the wavenumber domain from both the horizontal and vertical electric dipoles," in *Journal of Geophysics and Engineering*, vol. 13, no. 4, pp. 505-515, Jun. 2016.

[18] A. V. Grayver, R. Streich and O. Ritter, "Three-dimensional parallel distributed inversion of CSEM data using a direct forward solver,“ in *Geophysical Journal International*, vol. 193, no. 3, pp. 1432-1446, Jun. 2013.