# Letter

## A Novel Sensor Scheduling Algorithm Based on Deep Reinforcement Learning for Bearing-Only Target Tracking in UWSNs

Linyao Zheng, Meiqin Liu, *Senior Member, IEEE*, Senlin Zhang, *Member, IEEE*, and Jian Lan, *Senior Member, IEEE*

Dear Editor,

This letter is concerned with the energy-aware multiple sensor co-scheduling for bearing-only target tracking in the underwater wireless sensor networks (UWSNs). Considering the traditional methods facing with the problems of strong environment dependence and lack flexibility, a novel sensor scheduling algorithm based on the deep reinforcement learning is proposed. Firstly, the sensors' co-scheduling strategy in UWSNs is formulated as Markov decision process (MDP). Then, a dueling double deep Q network (D3QN) is developed to solve the MDP in a scalable and model free manner. Besides, the prioritized experience replay (PER) method is utilized to accelerate network convergence. Finally, the effectiveness and superiority of the proposed algorithm are confirmed by experimental results.

With the advantages of self-organization structure, low cost and strong concealment, UWSNs show a promising ability in underwater target passive tracking [1]. However, the battery-powered sensors in the UWSNs are hardly to be recharged in the depths of the ocean, severely limiting the lifetime of UWSNs. Therefore, it is essential to study an energy-efficient sensor co-scheduling strategy to make a tradeoff between tracking accuracy and energy consumption. In [2], a wake-up/sleep and valid measurement selecting method was proposed to increase the energy efficiency of the sensors in UWSNs. In [3], an adaptive sensor scheduling scheme was introduced, and energy can be saved by changing the sampling intervals according to tracking accuracy threshold at each time step. In [4], a novel underwater passive tracking framework in UWSNs based on dynamic clustering was proposed, scheduling the sensors by selecting cluster head and cluster members adaptively based on dynamic programming (DP) method. Although the above studies have already made good progress, the proposed methods greatly depend on environment and prior information, and lack flexibility in complex and dynamic underwater environments.

Compared with the traditional method, deep reinforcement learning (DRL) has no need for exactly prior knowledge of environment and has a strong ability to adapt the dynamic changes of the environment [5], which makes it more suitable for underwater environment. Besides, as demonstrated in [6], the DRL techniques can be effectively deployed in UWSNs.

Corresponding author: Meiqin Liu.

Citation: L. Y. Zheng, M. Q. Liu, S. L. Zhang, and J. Lan, "A novel sensor scheduling algorithm based on deep reinforcement learning for bearing-only target tracking in UWSNs," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 4, pp. 1077–1079, Apr. 2023.

L. Y. Zheng and J. Lan are with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: zhengLY@stu.xjtu.edu.cn; lanjian@xjtu.edu.cn).

M. Q. Liu is with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, and also with the State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China (e-mail: liumeiqin@zju.edu.cn).

S. L. Zhang is with the College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: slzhang@zju.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2023.123159

Motivated by the above discussions, in this letter, we aim to obtain an energy-efficient sensor scheduling policy for underwater passive tracking in UWSNs. To this end, following the underwater passive tracking framework in [4] and considering the characteristics of underwater passive tracking in UWSNs, we formulate the sensors' co-scheduling protocol as MDP. Then, the D3QN algorithm with PER is applied to obtain better learning performance. The main contributions of this letter are stated as follows: 1) The sensor co-scheduling strategy in UWSNs is formulated as MDP. 2) A mock data method is introduced to construct the reward function in the DRL environment to avoid the abuse of ground truth of target. 3) The D3QN algorithm with PER is introduced to solve the MDP to find a suitable schedule policy in a scalable and model-free manner.

**Problem statement:** 1) Underwater passive tracking framework: In this letter, the target motion model is assumed as constant velocity model (CVM) [7]. Referring to the underwater passive tracking framework in [4], there are $N_k$ cluster members (CM) and a cluster head $FC_k$to construct a dynamic cluster to participate in tracking at time $k$. Fig. 1 shows the basic idea of this framework. Furthermore, assuming the sensors in the UWSNs have the same communication range $R_c$ and the same sensing range $R_s$. Moreover, we define that all activated sensors make up the candidate cluster member set $E_k$ and the candidate cluster head set $F_k$ at time $k$. $E_k$ and $F_k$ satisfy the following conditions:

$$\begin{aligned} E_k &= \{P_i | I_i > I_{P\_th}, E_i \geq E_{P\_th}\} \\ F_k &= \{P_j | I_j > I_{FC\_th}, E_{\rm j} \geq E_{FC\_th}\} \end{aligned} \tag{1}$$

where $I_{P\_th}$ and $I_{FC\_th}$ are acoustic intensity thresholds of candidate cluster members and cluster heads respectively. $E_{th}^{P}$ and $E_{th}^{FC}$ are energy thresholds of candidate cluster members and cluster heads respectively.

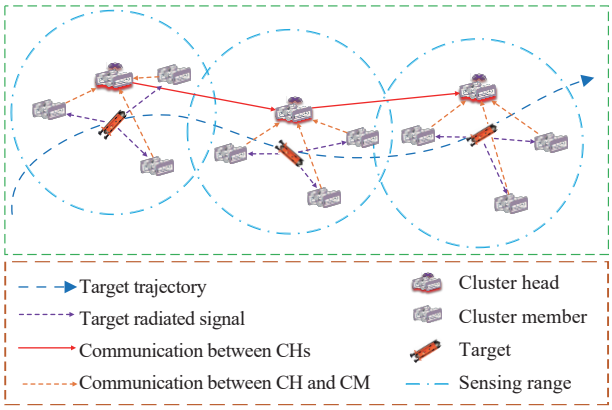


Fig. 1. The passive target tracking framework based on dynamic cluster.

2) Energy-efficient sensors' co-scheduling protocol: The objective of the co-scheduling in the above framework is to choose the suitable cluster members from the set $E_k$by cluster heads to make an optimal tradeoff between tracking accuracy and energy consumption.

The above objective is equivalent to maximize the objective function as follows:

$$\begin{aligned} J &= \mathop{\arg\max}_{P_k^* \in E_k, |P_k^*| = N_k^*} \psi(P_k^*) \\ \psi(P_k^*) &= \lambda \varphi_{\rm utility}(P_k^*) + (1-\lambda)\varphi_{\rm cost}(P_k^*) \end{aligned} \tag{2}$$

where $P_k^*$ is the sub set composed of $N_k^*$ cluster members which are chosen at time $k$, $\lambda$ is the joint factor which is used to balance the energy consumption and tracking accuracy, and $\varphi_{\rm utility}(\cdot)$ is the utility function representing the tracking performance, which is given by

$$\varphi_{\text{utility}}(P_k^*) = \det(\mathbf{J}_k^*),\ \ (P_k^* \in E_k, |P_k^*| = N^*) \tag{3}$$

here $\mathbf{J}_k^*$ is the fisher information matrix according to the positions of members in $P_k^*$ [4]. $\varphi_{\cos t}(\cdot)$ is the cost function representing the energy consumption, which is given by

$$\varphi_{\text{cost}}(P_k^*) = \sum_{i=1}^{N_k^*} \frac{E_{i,k}^{\text{initial}} - E_{i,k}^{p}}{E_{i,k}^{\text{initial}}},\ \ P_k^* \in E_k \tag{4}$$

where $E_{i,k}^{\text{initial}}$ is the initial energy of the $i$-th cluster member at time $k$, and $E_{i,k}^{p}$ is the energy consumption of the $i$-th cluster member.

**Proposed methods:** In this section, we shall introduce the detail of the proposed method from two aspects:

1) Formulation of MDP: Equation (4) can be formulated as an MDP, which is defined by a tuple $< S, A, R >$. Each element $< S, A, R >$ is defined as follows:

a) State space $S$: From the discussion above, the state of the MDP at time $k$ is given by

$$S^k = \{\psi(P_k^*) | P_k^* \in E_k\} \tag{5}$$

where the state $S^k$ is directly related to the objection function at time $k$, giving faster convergence for the algorithm [8].

b) Action space $A$: The action space corresponding to $\frac{N_k!}{N_k^*!(N_k - N_k^*)!}$ different ways of choosing the suitable cluster member from the set $E_k$ at time $k$. Therefore, we obtain

$$A_k = \{a_k^1, \ldots a_k^{N_k^*}\}. \tag{6}$$

c) Reward function $R$: The reward function includes the following two items: the current reward $r_k^c$ and the settlement reward $r^s$. The current reward at time $k$ is

$$r_k^c = \lambda \phi_{\text{utility}}(P_k^*) + (1-\lambda)\phi_{\cos t}(P_k^*),\ \ P_k^* \in E_k \tag{7}$$

which is utilized to maximize cumulative rewards.

The settlement reward $r^s$ is the huge reward representing each training result which can be reflected by tracking performance and system energy efficiency. However, the most of tracking performance evaluation methods assume exact knowledge of the ground truth, which is hard to be obtained in the practical underwater passive tracking. To solve this problem, we introduce the mock data method [9], which can evaluate the tracking performance by measuring the deviation between mock data generated by the estimate and the real measurement. Therefore, assuming that the time of a tracking is $T_d$, the tracking performance $\bar{d}_i$ of each training can be represented by

$$\bar{d}_i = \frac{\sum_{k=1}^{T_d} d_x^k(m_k^x, m_k^*)}{T_d} \tag{8}$$

where $m_k^*$ is the mock data and $m_k$ is the real measurement, $d_x^k$ is the Mahalanobis distance between the mock data and real measurement. The settlement reward is given by

$$r^s = [(d_{\text{goal}} - \bar{d}_i) \times \kappa] + [(E_{\text{goal}} - E_c^i) \times \mu],\ \ i = 1, 2, \ldots N_e \tag{9}$$

where $N_e$ is the number of training, $d_{\text{goal}}$ and $E_{\text{goal}}$ are the goal of tracking accuracy and energy consumption respectively, which are determined by the task requirements. $\kappa$ and $\mu$ are weighting factors, which are set for the tradeoff between the tracking performance and system energy consumption.

Finally, the reward function is expressed as

$$R = \begin{cases} r_k^c, & k < T_d \\ r^s, & k = T_d. \end{cases} \tag{10}$$

2) Solution by D3QN: In DRL, the key point of solving MDP is to obtain the expected return by maximizing state-action value $Q(s_k, a_k)$ which is approximated by the deep Q network. For better learning performance, we introduce the D3QN to solve the above MDP.

D3QN is composed of current network and target network which are deep Q networks with different parameters but the same structure, Here, current network and target network are composed of one input layer, two 128-layer full connection (FC) layers and one output layer. The parameter of current network is $\theta$ while the parameter of target network is $\theta'$. D3QN solves the MDP by updating the current network with loss function. The current network of D3QN at time $k$, is composed of value function and advantage function, which is expressed as

$$\begin{aligned} Q_k(s_k, a_k; \theta_k, p, q) = {} & V_k(s_k, a_k; \theta, q) \\ & + (A(s_k, a_k; \theta, p) - \frac{1}{N_A} \sum_{a_k^*} A(s_k, a_k^*; \theta, p)) \end{aligned} \tag{11}$$

where $a_k^*$ is all actions that can be taken at time $k$, $V_k(\cdot)$ is the value function, $A(\cdot)$ is the advantage function, $N_A$ is the number of actions, $p$ and $q$ are network parameters of value function and advantage function respectively.

To further improve the sampling efficiency and convergence speed, PER is employed to update the network parameter [10]. Then, the loss function is given by

$$L(\theta) = \frac{1}{N_m} \sum_{j=1}^{N_m} [(R + \gamma \max_{a'_k} Q_k(s_{k+1}, a'_k | \theta', p, q) - Q(s_k, a_k | \theta, p, q)]^2 \omega_j \tag{12}$$

where $s_{k+1}$ is the state at time $k$+1, and $a'_k$ is the action taken under the $s_{k+1}$. $\omega_j$ is the weight of priority sample, which is given by

$$\omega_j = \frac{1}{(N_r \cdot \vartheta_j)^{\beta} \max \omega_l} \tag{13}$$

where $N_r$ is the capacity of the replay buffer, and $\beta$ is the impact factor gradually increasing to 1.

In summary, the proposed D3QN-PER based sensor scheduling method is shown in Fig. 2.

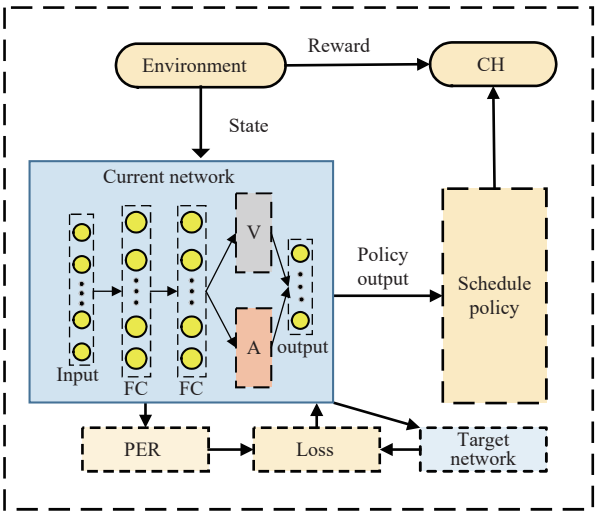


Fig. 2. The D3QN-PER based sensor scheduling method.

**Experiments:** A numerical example is provided to evaluate the performance of the proposed method in the underwater passive tracking scenario compared with some existing sensor schedule methods.

We consider the following existing methods:

1) The sensor schedule method based on DP in [4].

2)The sensor schedule method based on genetic algorithm (GA), which utilizes the GA method to solve the schedule problem in (6).

The initial settings of UWSNs and target are the same as those in [4]. The total observation time of system is 30 s. The number of cluster member $N_k^*$ is set as 3 and $|E_k| = 10$, meanwhile, the joint factor $\lambda$ of the objection function is set as 0.6. Overall, the simulation environment is shown in Fig. 3. The D3QN parameter setting is shown in Table 1, which is set by the rule in [5]. The training process of the proposed method is shown in Fig. 4. The GA method is imple-

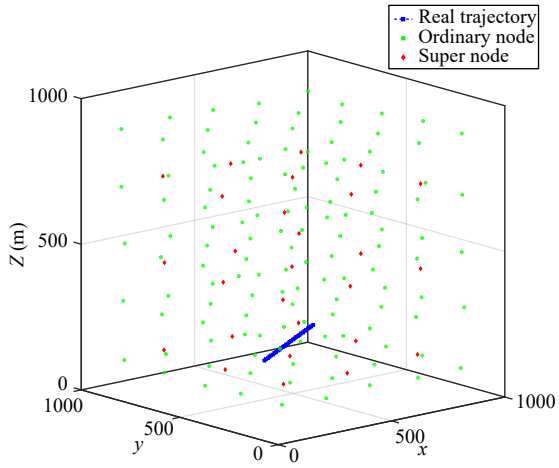


Fig. 3. The simulation environment.

Table 1. D3QN Parameter Setting

| Parameter | Value |
|---|---|
| Weighting factor $\kappa/\mu$ | 1500/2000 |
| $d_{\text{goal}}/E_{\text{goal}}$ | 0.8/0.23 |
| Minibatch $N_m$ | 128 |
| Replay buffer capacity $N_r$ | $10^5$ |
| Training episode $N_e$ | 200 |
| Network update frequency $f_u$ | 30 |
| PER parameter $\beta$ | 0.4 |
| Learning rate | 0.000 25 |
| Discount factor $\gamma$ | 0.93 |
| Activation function | ReLU |
| Exploration probability $\varepsilon$ | 0.017 |

mented by the GA tools in Python [11].

Our experiment uses an AMD Core 5800X CPU @ 3.80 GHz, NVIDIA GeForce RTX3080 GPU, and Windows 1064 bit. We use Python 3.8 and Pytorch 1.11.0 to realize the proposed method.

To access the target passive tracking accuracy, the root mean square error (RMSE) is adopted to evaluate the performance of our algorithm. The RMSE data of these compared methods in 100 Monte Carlo tests is shown in Fig. 5. Furthermore, to evaluate the energy consumption, we record the energy consumption in Fig. 5.

As shown in Fig. 4, After around the 130th episode, the reward keeps stable high scores, which illustrates the convergence of the proposed algorithm. In Fig. 5, the RMSE result illustrates that the tracking accuracy of D3QN-PER based method is better than that of

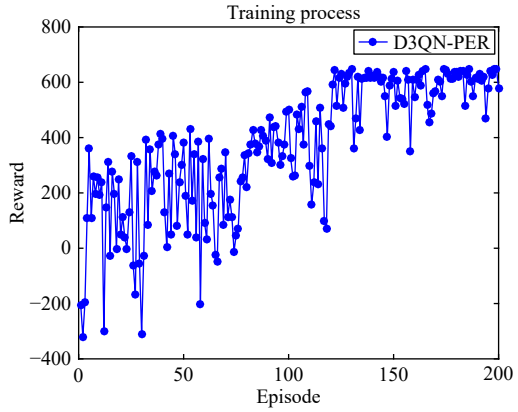


Fig. 4. The training process.

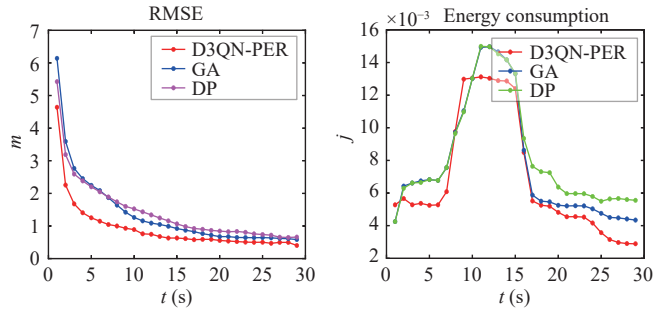


Fig. 5. RMSE and energy consumption.

DP based method and GA based method. Besides, Fig. 5 also shows that the D3QN-PER based method has lower energy consumption compared with other methods. It is seen from Fig. 5 that the proposed method performs better than the methods compared.

**Conclusions:** This letter has proposed a new DRL-based sensor schedule method for underwater passive tracking in UWSNs. The schedule problem is formulated as MDP and a mock data method is introduced to construct the reward function to avoid the abuse of ground truth of target. Furthermore, the D3QN-PER algorithm is introduced to solve the MDP to find a suitable schedule policy in a scalable and model-free manner. Finally, the simulation results confirm the effectiveness and the superiority of the proposed method.

**Acknowledgments:** This work was supported by the National Natural Science Foundation of China (62173299, U1809202), the Joint Fund of Ministry of Education for Pre-Research of Equipment (8091B022147), and the Fundamental Research Funds for the Central Universities (072022001).

## References

[1] J. Luo, H. Ying, and L. Fan, "Underwater acoustic target tracking: A review," *Sensors*, vol. 18, no. 1, p. 112, 2018.

[2] C. H. Yu, J. C. Lee, J. W. Choi, M.-K. Park, and D. J. Kang, "Energy efficient distributed interacting multiple model filter in UWSNs," in *Proc. 12th Int. Conf. Control, Automation and Syst.*, 2012, pp. 1093–1098.

[3] S. Zhang, H. Chen, and M. Liu, "Adaptive sensor scheduling for target tracking in underwater wireless sensor networks," in *Proc. Int. Conf. Mechatronics Control*, 2014, pp. 55–60.

[4] X. Han, M. Liu, S. Zhang, and Q. Zhang, "A multi-node cooperative bearing-only target passive tracking algorithm via UWSNs," *IEEE Sensors J.*, vol. 19, no. 22, pp. 10609–10623, 2019.

[5] A. Feriani and E. Hossain, "Single and multi-agent deep reinforcement learning for AI-enabled wireless networks: A tutorial," *IEEE Commu. Surveys & Tutorials*, vol. 23, no. 2, pp. 1226–1252, 2021

[6] R. Su, Z. Gong, D. Zhang, C. Li, Y. Chen, and R. Venkatesan, "An adaptive asynchronous wake-up scheme for underwater acoustic sensor networks using deep reinforcement learning," *IEEE Trans. Vehicular Technology*, vol. 70, no. 2, pp. 1851–1865, 2021.

[7] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. Part I. Dynamic models," *IEEE Trans. Aerospace and Electronic Syst.*, vol. 39, no. 4, pp. 1333–1364, 2003.

[8] X. Leong, A. S. Ramaswamy, A. Quevedo, and D. E. Karl, "Deep reinforcement learning for wireless sensor scheduling in cyber–physical systems," *Automatica*, vol. 113, p. 108759, 2020.

[9] W. Cao, J. Lan, and X. R. Li, "Joint tracking and classification based on recursive joint decision and estimation using multi-sensor data," in *Proc. 17th Intern. Conf. Information Fusion*, 2014, pp. 1–8.

[10] H. Song, Y. Liu, J. Zhao, J. Liu, and G. Wu, "Prioritized replay dueling DDQN based grid-edge control of community energy storage system," *IEEE Trans. Smart Grid*, vol. 12, no. 6, pp. 4950–4961, 2021.

[11] W. Lee and H. Y. Kim, "Genetic algorithm implementation in python," in *Proc. 4th Annual ACIS International Conf. Computer and Infor. Science*, 2005, pp. 8–11.