Complex & Intelligent Systems (2025) 11:207
https://doi.org/10.1007/s40747-025-01788-x

ORIGINAL ARTICLE

Target pursuit for multi-AUV system: zero-sum stochastic game with WoLF-PHC assisted

Le Hong[1,2,3] · Weicheng Cui[2,3]

Received: 3 July 2024 / Accepted: 24 January 2025

Abstract

Due to the complexity of the underwater environment and the difficulty of the underwater energy recharging, utilizing multiple autonomous underwater vehicles (AUVs) to pursue the invading vehicle is a challenging project. This paper focuses on devising the rational and energy-efficient pursuit motion for a multi-AUV system in an unknown three-dimensional environment. Firstly, the pursuit system model is constructed on the two-player zero-sum stochastic game (ZSSG) framework. This framework enables the fictitious play on the behaviors of the invading AUV. Fictitious play involves players updating their strategies by observing and inferring the actions of others under incomplete information. Under this framework, a relay-pursuit mechanism is adopted by the pursuit system to form the action set in an energy-efficient way. Then, to reflect the pursuit goals of capturing the invading vehicle as soon as possible and avoid it from reaching its point of attack, two corresponding pursuit factors are considered in the designed reward function. To enable the pursuit AUVs to navigate in an unknown environment, WoLF-PHC algorithm is introduced and applied to the proposed ZSSG-based model. Finally, simulations demonstrate the effectiveness, the advantages, and the robustness of the proposed approach.

Keywords Autonomous underwater vehicles · Zero-sum stochastic game · Fictitious play · Relay-pursuit · WoLF-PHC

Introduction

Autonomous underwater vehicles (AUVs) have undergone significant advancements, making them highly autonomous [1]. This leads to an inevitable trend of utilizing multiple AUVs to conduct the underwater protection and defense [2]. To facilitate the target pursuit for the multi-AUV system, there is an urgent need for (1) developing a pursuit system model capable of analyzing the behaviors of all vehicles in the underwater three-dimensional environment, and (2) proposing a specific methodology to determine the effective and efficient action for the pursuit system in the unknown environment.

In the field of multi-robot target pursuit, game theory is highly compatible with the goal opposition, non-cooperation relationship, and action dependence [3]. Traditional target pursuit games are differential games, fail to capture the dynamic, uncertain, and competitive nature of real-world underwater environments. Considering the real-world perturbations and parameter switching, the mainstream game model has gradually shifted towards the stochastic game (SG). It is because that for complex and uncertain systems, dynamic decision-making and control are crucial [4–6]. In this regard, the state transition process of the SG, as a Markov dynamic process, provides valuable insights for determining dynamic strategies. It involves a class of dynamic games with state transfer probabilities and a series of stages.

Compared with the other SG models, the two-player zero-sum stochastic game (ZSSG) is well-suited to the target pursuit problem. It naturally captures the competitive and adversarial interactions between the vehicles and the target. Traditional differential games, while effective in certain deterministic environments, lack the ability to handle the stochastic and dynamic nature of real-world underwater

✉ Weicheng Cui
cuiweicheng@westlake.edu.cn

Le Hong
hongle@westlake.edu.cn

1 Zhejiang University-Westlake University Joint Training, Zhejiang University, Hangzhou 310024, Zhejiang, China

2 Key Laboratory of Coastal Environment and Resources of Zhejiang Province (KLaCER), School of Engineering, Westlake University, Hangzhou 310024, Zhejiang, China

3 Institute of Advanced Technology, Westlake Institute for Advanced Study, Hangzhou 310024, Zhejiang, China

Published online: 13 March 2025

environments [7]. In contrast, the ZSSG framework has the probabilistic transitions and the inherent uncertainty in the environment, enabling a more robust and realistic modeling of the pursuit-evasion dynamics [8, 9]. Moreover, the ZSSG model accounts for the zero-sum nature of the target pursuit problem, where the gain of one player is the loss of the other. It also ensures the existence of a Nash equilibrium, which guarantees stable and effective strategy optimization [10]. At last, by leveraging multi-agent reinforcement learning (MARL) techniques within this framework, vehicles can continuously adapt and refine their pursuit strategies against the target. They learn from their interactions through fictitious play and the experience of adjusting to the uncertain environment [11]. Integrating MARL into the ZSSG model allows vehicles to respond to the uncertain environment. It also allows the vehicles to improve their strategies over time, making this approach applicable for real-world applications.

Despite significant advances in MARL algorithms, several theoretical and technical challenges remain when applying them to underwater target pursuit within the ZSSG framework. First, many MARL algorithms require substantial computational resources. This challenge limits their practical applicability in real-world scenarios, particularly in resource-constrained environments. Among these MARL algorithms, the WoLF-PHC algorithm stands out due to its lower computational complexity. This characteristic makes it more practical for deployment in environments with limited computational power [12]. Second, human-like rationality is crucial for decision-making in underwater target pursuit. The involved pursuit system needs to make reasonable decisions under incomplete information and uncertain conditions. This paper addresses this challenge by using the WoLF-PHC algorithm. The WoLF-PHC approach is helpful for agents to make sound decisions in uncertain and complex environments [13, 14]. For example, in cloud computing resource allocation, multiple virtual machines engage in a strategic interaction similar to a game. Virtual machines adjust their resource allocation strategies by adapting their learning rates based on past experiences [15]. This process reflects human-like rationality by dynamically balancing exploration and exploitation—learning faster when underperforming and slowing down when nearing optimal performance, similar to how humans adjust their decisions in uncertain and complex environments. Finally, it is important that the pursuit system model and simulations reflect real-world conditions when deploying multi-AUV systems. This is particularly crucial in MARL-assisted algorithms, as agents learn from interactions in dynamic and uncertain environments. If the model does not account for these complexities, the agents may not perform well in real-world scenarios. In this paper, we incorporate real-world AUVs into the simulation. This ensures that the MARL-assisted system model captures underwater conditions and operational constraints. As a result, the proposed underwater pursuit model could be more accurate and better suited for real-world applications.

Table 1 Terminology

Abbreviation	Full name
AUV	Autonomous underwater vehicle
SG	Stochastic game
ZSSG	Zero-sum stochastic game
MARL	Multi-agent reinforcement learning
WoLF	Win or learn fast
PHC	Policy-hill climbing
HMI	Human–machine interface
USV	Unmanned surface vehicle
RL	Reinforcement learning
A2C	Advantage actor critic
PPO	Proximal policy optimization
REMUS	Remote environmental monitoring units

In light of the above discussion, the main contributions of this paper are as follows:

1. Design of a three-dimensional ZSSG-based multi-AUV pursuit model with specific action-sets and reward function: in the established ZSSG-based system model, a multi-AUV pursuitsystem could fictitiously play the evasion actions of the invading AUV, thereby enhancing its decision-making ability.
2. Application of a rational and convergent MARL algorithm: WoLF-PHC algorithm is employed to assist the optimal action determination in the proposed ZSSG-based model. This integration enables the pursuit system to navigate in an uncertain environment while also improving computational efficiency.
3. Simulating a multi-AUV pursuitsystem as closely to the reality as possible: a real-world AUV, known for its exceptional capabilities, is used to inform the parameter setting and deployment in the pursuit system. This helps improve the realism of the simulation environment by ensuring more accurate underwater pursuit scenarios.

The remainder of the paper is organized as follows: "Related work" reviews related work. "System model" introduces the ZSSG-based system model developed for multi-AUV target pursuit. "Methodology" details the proposed methodology, with a focus on the WoLF-PHC algorithm within the ZSSG framework. "Simulations" presents the simulation results, and "Conclusions" concludes the findings. To facilitate understanding of key concepts, Table 1 lists the main terminology along with their definitions.

Related work

In this section, we review relevant literature on game-based methods for the multi-agent target pursuit, the specific MARL methods designed for the ZSSGs, and the adopted MARL techniques in the marine pursuit.

Games for multi-agent target pursuit

Three popular games for modeling the multi-agent system to pursue a target are the differential games [16], matrix games [17], and the SGs [18]. Since the key to solving the differential target pursuit games is to solve the nonlinear partial differential Hamilton–Jacobi–Isaacs equation, which lacks an analytical solution in most cases [19]. Matrix games, while extensively discussed, they still depend on the known setting of the matrix payoff formulation [20]. In this context, SGs, a class of dynamic games with state transfer probabilities, have gradually become the mainstream of target pursuit games [21]. As the state transfer process in SGs follows a Markov process, SGs are also suitable for applying reinforcement learning (RL) to improve its adaptability and robustness in an uncertain environment [22]. In addition, for both finite-stage and finite-state SGs, there exist Nash equilibrium solutions for them [10].

Among the range of SG models, the two-player ZSSG is particularly effective for target pursuit. It captures the competitive and adversarial interactions between the pursuit agent and the invading target. In a ZSSG, the gain of one agent corresponds directly to the loss of the other, accurately reflecting the zero-sum nature of the target pursuit, where the success of the pursuer comes at the expense of the evader. Moreover, the existence of a Nash equilibrium in ZSSG ensures that both players can develop optimal strategies, providing a stable solution where neither player has an incentive to change their strategy. This enables the pursuit agent to fictitiously play the actions of the target, increasing the chances of success even in the uncertain environments [23]. Given these advantages, ZSSG proves to be a powerful model for capturing both the competitive and uncertain aspects of multi-agent target pursuit. Compared with other game models and pursuit methods, it offers a more realistic and adaptable approach for the target pursuit.

MARL methods for ZSSGs

Since ZSSG establishes an adversarial, zero-sum structure for the target pursuit, MARL algorithms are helpful for optimizing strategies within the uncertain environments of the ZSSG-based framework. Specifically, MARL is effective for handling complex multi-agent interactions. In multi-agent target pursuit scenarios, the MARL algorithm usually controls all agents in a centralized way [18, 24]. The goal is to achieve an equilibrium point using a limited number of interaction samples. Since the process of seeking the worst-case optimality for each player in the ZSSG-based model can be treated as solving a Markov dynamic process [20], dynamic programming methods such as least-squares policy iteration and neural fitted Q iteration can be adopted to solve ZSSGs [25, 26]. Under these basis, policy-based MARL approaches can also be applied. To be specific, a practical solution named minimax-Q is proposed to replace the max operator with the minimax operator in the ZSSG-based frameworks [27, 28]. Besides, the asymptotic convergence results of the minimax-Q approach are developed in both tabular cases and value function approximations [29]. To avoid the overly pessimism property by playing the minimax value, WoLF was proposed to take steps to exploit the sub-optimal policy of the opponent for a higher reward on a variety of SGs [30]. Then, WoLF is further generalized in multi-player zero-sum repeated games [31].

On the other hand, in the target pursuit, the most important factor is the real-time implementation for providing the optimal decisions. Therefore, while the state-of-the-art MARL algorithm behaves well in multi-agent system under the large-model, they are often not suitable for real-time decision-making due to their high computational complexity and long training times [32]. In contrast, lightweight MARL methods, such as the minimax Q-learning, or WoLF, are better suited for real-time decision-making as they are more efficient and can adapt quickly to dynamic environments. Furthermore, to enhance decision-making rationality, we combine WoLF with PHC to support the ZSSG-based pursuit system. Like Q-learning, the PHC method is rational and converges to an optimal policy. In this combined approach, WoLF-PHC retains its rationality, with only the learning rate being adjusted [15].

MARL methods in the marine pursuit

The initial phase of marine pursuit explores the use of multiple unmanned surface vehicles (USVs) to carry out target pursuit tasks. Specifically, it focuses on enabling multiple USVs to engage in self-organizing cooperative pursuit movements. This takes place in a two-dimensional open water environment, with the goal of capturing an intelligent evader [33]. Then, to improve the robustness of USVs in a multi-obstacle environment, a target pursuit approach is introduced. This approach incorporates a deep RL method to train the pursuit model of the pursuer. It is also combined with the imitation learning to develop the escape model for the evader [34]. Furthermore, to reflect the competition between the pursuit system and the invading USV, a multi-USV pursuit system model is constructed on a zero-sum game-based framework [35]. In this framework, the min–max Q-learning is adopted to approximate the payoff function during the

pursuit [28]. The above approaches show promises for multi-robot system in marine pursuit field. However, developing effective target pursuit methods for multiple AUVs in the uncertain three-dimensional underwater environments is still an unresolved challenge. The challenge remains open for further exploration.

System model

Preliminaries for ZSSG

SG, known as Markov game, has been widely used as a canonical model for dynamic multi-agent interactions [36, 37]. At each time step, $k = 0, 1, \cdots$, players play a stage game that corresponds to a particular state in a multi-state environment. The state of the SG evolves stochastically. This evolution is determined by the transition probabilities, which depend on the joint actions of all players. Specifically, a two-player ZSSG is characterized by a tuple $SG := \langle S, A := \mathcal{A}_1 \times \mathcal{A}_2, r, p, \gamma \rangle$. S denotes the finite set of states. $\mathcal{A}_1$ and $\mathcal{A}_2$ represent the finite action sets that the player 1 and the player 2 can take at any state, respectively. Then, the joint action set could be defined by $A = \mathcal{A}_1 \times \mathcal{A}_2$.

As the ZSSG contains zero-sum game, the reward functions for the two players are opponent. When the two players at a state $s \in S$ and play the joint action vector $a \in A$, the player 1 will get the reward $r(s, a)$, while the player 2 will get the opponent reward $-r(s, a)$. The transition probability from state s to $s\prime$ in the next time step is $p(s\prime|s, a)$. Besides, the two players also discount the impact of future payoff in their rewards with the discount factor $\gamma \in [0,1)$. Therefore, the objective of player 1 is to maximize the expected sum of discounted stage-payoffs collected over infinite horizon, given by:

$$\mathbb{E}\left\{\sum_{k=0}^{\infty} \gamma^k r(s_k, a_k)\right\}, \tag{1}$$

where $a_k \in A$ denotes the action vector played by the two players at the time step k. The objective for the player 2 is the opponent of the Eq. (1).

The two players can choose an infinite sequence of mixed actions. If players have full knowledge of all their previous actions and observations, they can mix their actions independently based on their behavioral strategy.

Definition 1 (*Behavioral Strategy*) A behavioral strategy of the player 1 is $\pi^1 : S \to \Delta(\mathcal{A}_1)$. For all $s \in S$, $\pi^1(s)$ is a probability distribution on $\mathcal{A}_1$.

Definition 2 (*Value Function*) According to the definition 1 and the objective of player 1, the behavioral strategy profile of the two-player ZSSG is defined by $\pi := \{\pi_1, \pi_2\}$. Therefore, a value function of player 1 under the strategy profile π is formed as follows:

$$V_1^{\pi}(s) = \mathbb{E}\left\{\sum_{k=0}^{\infty} \gamma^k r(s_k, a_k)\right\}, \tag{2}$$

where $\{s_0, s_1, \cdots\}$ is the Markov chain such that the transition matrix is p^{π}, $p^{\pi}(s'|s) = p(s'|s, a)$ for all $k = 0,1, \cdots$. Therefore, the value function can be further formulated as follows via the Bellman policy equation:

$$V_1^{\pi}(s) = \mathbb{E}_{a\sim\pi(s)}\left[r(s, a) + \gamma \sum_{s'\in S} p(s'|s, a) V_1^{\pi}(s')\right]. \tag{3}$$

To calculate the equilibrium in the ZSSG is to calculate the sequential stationary Nash equilibrium in the Markov chain. The definition and the existence of it in the ZSSG are illustrated as follows.

Definition 3 (*Stationary Nash equilibrium*) If the following equations hold with $\varepsilon = 0$, a stationary strategy profile π is defined as a stationary mixed-strategy Nash equilibrium at state s.

$$V_1^{\pi_1,\pi_2}(s) \geq V_1^{\widetilde{\pi}_1,\pi_2}(s) - \varepsilon for all \widetilde{\pi}_1, \tag{4}$$

$$V_2^{\pi_1,\pi_2}(s) \geq V_2^{\pi_1,\widetilde{\pi}_2}(s) - \varepsilon for all \widetilde{\pi}_2. \tag{5}$$

Theorem 1 (Existence of the stationary Nash equilibrium in SGs [38]) *As for the SGs with finite players, states and actions, and discount factor $\gamma \in [0,1)$, a stationary mixed-strategy equilibrium always exists.*

ZSSG-based system model

In this paper, the proposed ZSSG-based system model $SG := \langle S, A := \mathcal{A}_p \times \mathcal{A}_e, r, p, \gamma \rangle$ contains two players: the pursuer, which contains a pursuit team of n AUVs, and the evader, i.e., the invading AUV. The corresponding framework is shown in Fig. 1. The pursuit system has two goals of capturing the invading AUV within its survival time and avoid the invading vehicle from reaching its point of attack.

As presented in Fig. 1, the decision-making process is as follows: firstly, the two players obtain the state s from the environment, i.e., the location-profile of all vehicles, denoted by $s := \{\boldsymbol{x_p}, \boldsymbol{x_e}\}$. $\boldsymbol{x_p}$ and $\boldsymbol{x_e}$ is the three-dimensional location-profile of all the pursuit AUVs and the invading AUV, respectively. Then, once the state $s := \{\boldsymbol{x_p}, \boldsymbol{x_e}\}$ is determined as the position information of all vehicles, the next step for both players is to select their actions. The two players would choose the actions a_p and a_e from their action-sets $\mathcal{A}_p$ and $\mathcal{A}_e$, respectively. Based on the taken actions, the two players would receive the corresponding rewards, i.e., $r(s, a)$ and $-r(s, a)$, where $a \in A : A := \mathcal{A}_p \times \mathcal{A}_e$. At last, the multi-AUV pursuit system repeats the above three

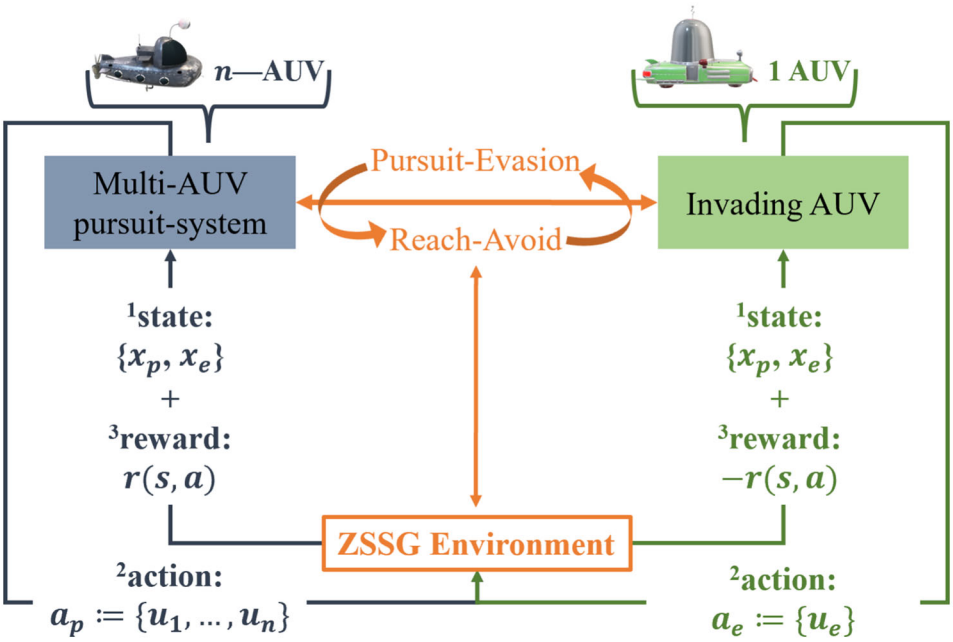


Fig. 1 ZSSG-based framework in the target pursuit for the multi-AUV system

steps to get the sequential stationary Nash equilibrium for the continuous decision-making.

To ensure practical implementation and computational feasibility, the continuous decision-making process is discretized. This discretization allows the decision-making model to represent continuous movements while maintaining computational efficiency. To be specific, in the proposed model, the location space is discretized by dividing the three-dimensional environment into a grid of finite-sized cells. Let $\boldsymbol{x}(t_k)$ represent the continuous position of the vehicle at time t_k. The three-dimensional space is then discretized into uniform grid cells, each with a fixed size Δx, Δy, and Δz along the respective dimensions. The discretized position $\boldsymbol{x}_d(t_k)$ of the vehicle at time t_k can be approximated as:

$$\boldsymbol{x}_d(t_k) = \left(\left\lfloor \frac{x(t_k)}{\Delta x} \right\rfloor \Delta x, \left\lfloor \frac{y(t_k)}{\Delta y} \right\rfloor \Delta y, \left\lfloor \frac{z(t_k)}{\Delta z} \right\rfloor \Delta z \right), \tag{6}$$

where $\lfloor \cdot \rfloor$ denotes the floor function, which rounds the value inside the brackets down to the nearest integer. Based on the spatial resolution Δx, Δy, and Δz, this operation maps the continuous location values $x(t_k)$, $y(t_k)$, and $z(t_k)$ to their corresponding discrete grid points.

For simpler discretization of movement, the time intervals between location updates can be used. The total time is divided into discrete time steps, Δt. The location of the vehicle is then updated at each time step as follows:

$$x(k+1) = x(k) + f(\Delta t), \tag{7}$$

where $f(\Delta t)$ represents a function of the discrete time step. This function updates the location based on the kinematic model of the vehicle. Equation (7) defines the movement rules. It ensures that the discretized location updates comply with the physical constraints of vehicle movement.

Based on the above discretization process, the discretization scale has impact on vehicle movement. A larger discretization scale improves computational efficiency but reduces decision-making accuracy and increases response time in dynamic tasks. On the other hand, a smaller discretization scale enhances accuracy but increases the computational burden. The high computational burden makes it unsuitable for real-time tasks. Therefore, in the simulation experiments, we selected a moderate discretization scale Δt. This choice balances accuracy and computational efficiency, meeting the needs of practical applications. All in all, the state s has been determined to be the location information of all the vehicles at the corresponding discretized time step. Therefore, the next step is to design the other two components for the proposed model: the action-sets for the two players, and the reward function for the pursuit system.

Action-set

To design a feasible action-set, the kinematic model for the vehicle in the three-dimensional environment needs to be constructed. First, several symbols are introduced. $\boldsymbol{x_i} = [x_i, y_i, z_i]^T$ and $\boldsymbol{x_e} = [x_e, y_e, z_e]^T$ denote the positions of the ith pursuit AUV and the invading AUV, respectively. $v_i(k)$ and $v_e(k)$ represent the velocities at the kth time step. These velocities can vary at each time step to account for different types of motion, such as uniform motion, acceleration, and deceleration. To simplify the experimental process, the

Table 2 Notations

Parameter	Full name
i	Index of the pursuit AUV
n	Total number of the pursuit AUV
e	Invading AUV
k	Current time step
$\mathcal{A}_p$	Action-set for the pursuit system
$\mathcal{A}_e$	Action-set for the invading AUV
a_p	Unit action-profile for the pursuit system
a_e	Unit action for the invading AUV
x(m)	Position in the x-coordinate
y(m)	Position in the y-coordinate
z(m)	Position in the z-coordinate
$\boldsymbol{x}$	Three-dimensional position
Δt(s)	Discrete time period
v(m/s)	Scalar velocity
u_x(m/s)	Unit velocity in the x-coordinate
u_y(m/s)	Unit velocity in the y-coordinate
u_z(m/s)	Unit velocity in the x-coordinate
$\boldsymbol{u}$	Three-dimensional unit velocity (motion)
l(m)	Length of the AUV

motion of the vehicles is assumed to be uniform in the simulation. Δt is the time period between the two time steps. $\boldsymbol{u_i} = \left[u_{x_i}, u_{y_i}, u_{z_i}\right]^T$ and $\boldsymbol{u_e} = \left[u_{x_e}, u_{y_e}, u_{z_e}\right]^T$ are the unit vectors of the velocity for the corresponding vehicles. These vectors control the actions of the vehicles in the proposed model. Therefore, at each time step, $\|\boldsymbol{u_i}\| = \|\boldsymbol{u_e}\| = 1$. As a result, the unit actions for the two players are designed as: $a_p := \{\boldsymbol{u}_1, \ldots, \boldsymbol{u_n}\}$ and $a_e := \{\boldsymbol{u_e}\}$. The notations for the above symbols are also presented in Table 2.

Based on these introduced parameters, it is assumed that all the AUVs are in the uniform motion. Thus, based on Eqs. (6) and (7), at the kth time step, the kinematic model for the two types of vehicles can be expressed as follows:

$$\boldsymbol{x_i}(k+1) = \boldsymbol{x_i}(k) + v_i(k)\boldsymbol{u_i}(k)\Delta t, \tag{8}$$

$$\boldsymbol{x_e}(k+1) = \boldsymbol{x_e}(k) + v_e(k)\boldsymbol{u_e}(k)\Delta t, \tag{9}$$

where the controlled vectors $\boldsymbol{u_i}$ and $\boldsymbol{u_e}$ are assumed to have M and N selections, respectively. Then, the calculation dimension for obtaining the equilibrium at each time step is $M^n N$. This calculation dimension is so large that a huge amount of resources would be cost to achieve the equilibrium. Therefore, a relay-pursuit mechanism is adopted for the pursuit team to design a simpler action-set $\mathcal{A}_p$. As a result, the calculation dimension could be reduced to MN. The definition for this relay-pursuit mechanism is shown as follows:

Definition 4 (*Relay-Pursuit Mechanism*) Only one pursuit AUV is active, while the others are stationary. The determination for the active pursuit AUV changes over time, which depends on the outcome of each game. At a certain time step, if the active vehicle has been determined by the index i^*, the unit velocity $\boldsymbol{u_i}$ for the i^{th} pursuit AUV could be obtained as follows:

$$\boldsymbol{u_i} = \begin{cases} \frac{\boldsymbol{x_e} - \boldsymbol{x_i}}{\|\boldsymbol{x_e} - \boldsymbol{x_i}\|}, if\, i = i^* \\ 0, others \end{cases}. \tag{10}$$

Under the relay-pursuit mechanism, the pursuit team has total n actions. When $i \in \{1,2, \cdots, n\}$, the ith action for the pursuit team denotes that the i^{th} AUV is active. Similarly, the invading AUV's restricted action-set is designed to include $n+1$ choices. When $j \in \{1, \cdots, n\}$, the jth action for the invading vehicle is evading the jth pursuit AUV. The $n+1$ action is the target-seeking behavior of the invading vehicle, which means that it moves toward its point of attack $\boldsymbol{x_a}$.

Reward function

Since the reward functions for the two players are opposite in the ZSSG-based model, only the reward function for the multi-AUV pursuit system needs to be formulated. The reward function for the proposed pursuit model should reflect two objectives: (i) capturing the invading vehicle as soon as possible within the survival time t_s, and (ii) preventing the invading vehicle from reaching its point of attack $\boldsymbol{x_a}$. In this regard, the entries for the above two goals are assumed by the minimum time that the team would take to capture the invading vehicle, and the extent to which invading vehicle's heading is towards the target from the view of its current location, respectively.

To quantify the first goal, several time metrics are introduced here [17]. They are all linked to the minimum positive solution for t in the following equation:

$$\left(v_e^2 - v_i^2\right)t^2 + 2\left(v_e \boldsymbol{u_e}^T \boldsymbol{r_i} - l v_i\right)t + \|\boldsymbol{r_i}\|^2 - l^2 = 0, \tag{11}$$

where $\boldsymbol{r_i} = \boldsymbol{x_e} - \boldsymbol{x_i}$. l is the length of the pursuit vehicle. Once $\boldsymbol{u_e}$ is determined, the corresponding time $t(\boldsymbol{x_e}, \boldsymbol{x_i}, \boldsymbol{u_e})$ can be obtained, representing the minimum time for the i^{th} AUV to capture the invading vehicle. Therefore, the first component of the reward function is formulated by:

$$r_1(s, a) = -t(\boldsymbol{x_e}, \boldsymbol{x_i}, \boldsymbol{u_e}). \tag{12}$$

In terms of the second component of the reward function, it is defined by the extent to which invading vehicle's heading is towards from its current location. Therefore, it is assumed to be the cosine of θ, where θ is the angle between the vectors

$\boldsymbol{u_e}$ and $\boldsymbol{x_a} - \boldsymbol{x_e}$. When the invading vehicle chooses the j^{th} action and the pursuit system activate the i^{th} AUV, the second component of the reward could be given by:

$$r_2(s, a) = -\cos(\angle(\boldsymbol{u_e} - (\boldsymbol{x_a} - \boldsymbol{x_e}))), \tag{13}$$

where $\angle(\boldsymbol{u_e} - (\boldsymbol{x_a} - \boldsymbol{x_e}))$ denotes the angle between the $\boldsymbol{u_e}$ and the vector $\boldsymbol{x_a} - \boldsymbol{x_e}$.

Finally, under the state $s := \{\boldsymbol{x_p}, \boldsymbol{x_e}\}$ and the joint action $a := \{a_{p_i}, a_{e_j}\}$, the overall reward function for the pursuit system is formulated as follows:

$$r(s, a) = \frac{r_1}{\max\limits_a r_1} + r_2, \tag{14}$$

where r_1 is normalized with its maximum value. This ensures that all values of $\frac{r_1}{\max\limits_a r_1}$ are uniformed between zero and one so that they could be similar to the values in r_2.

Methodology

To enable the AUVs to navigate effectively and efficiently in an uncertain environment, WoLF-PHC is selected. This choice helps the constructed system model by reducing its dependency on the parameter setting and the formulation of the goal function [14, 39].

Learning in the ZSSG

Learning in the ZSSG-based model can be viewed as a model-free version of the value iteration in the Markov decision process. The corresponding update rule is given by:

$$\widehat{q}_{k+1}(s, a) = \widehat{q}_k(s, a) + \gamma\left(r_k + \beta \max_{\widetilde{a} \in A} \widehat{q}_k(\widetilde{s}, \widetilde{a}) - \widehat{q}_k(\widetilde{s}, \widetilde{a})\right), \tag{15}$$

where the triple $< s, a, \widetilde{s} >$ denotes respectively the current state s, current actions a, and the next state $\widetilde{s}$. The payoff r_k corresponds to the payoff received, i.e., $r_k = r(s, a)$. $\gamma \in [0,1)$ is the discount factor specific to the state-action pair (s, a). β is the learning rate set for the update rule. Besides, the Q-values for the same state-action pairs do not get updated, i.e., $\widehat{q}_{k+1}(s\prime, a\prime) = \widehat{q}_k(s\prime, a\prime)$. The Q-learning process in the ZSSG-based model guarantees almost certain convergence, as established by a rigorous proof [40].

WoLF-PHC-assisted algorithm

Based on the introduced learning process, Fig. 2 illustrates the application of the WoLF-PHC algorithm in the constructed system model. During the learning process, to maximize the expected reward, the two players continually learn and adapt their behavioral strategies, π_p and π_e. To update π_p and π_e, the WoLF-PHC adopts two learning rates δ_w and δ_l, where $\delta_w < \delta_l$. This means that the behavioral strategy would be updated slowly while winning, and be updated quickly while losing [41]. To determine the winning or loss, a baseline is designed. The baseline is the expected reward under the average probability of the behavioral strategies, denoted by $\overline{\pi}_p$ and $\overline{\pi}_e$.

According to the definition of the Nash equilibrium for the ZSSG-based model in Definition 3, the process for using WoLF-PHC to obtain the optimal action $a_p(k)$ and $a_e(k)$ is outlined. The detailed steps are presented in Algorithm 1. $a_{p_{i^*}}$ denotes that the i^*th AUV is chosen to be active. $a_{e_{j^*}}$ represents that the invading AUV adopts thej^*th action.

Algorithm 1 WoLF-PHC for the proposed model

Input: $i \in \{1,\cdots,n\}$:$\boldsymbol{x_i}(0)$,v_i; $\boldsymbol{x_e}(0)$;v_e; γ;δ_w; δ_l;
Output: Sequential a_p and a_e;
while $k \leq K$ **do**
  update $\boldsymbol{x_i}(k)$, $\boldsymbol{x_e}(k)$, and $\hat{q}_k(s,a)$;
  obtain the optimal behavioral strategies π_p and π_e;
  update $\overline{\pi}_p$ and $\overline{\pi}_e$;
  for $a_{p_i} \in \mathcal{A}_p$ **do**
    $\overline{\pi}_p(k)_i = \overline{\pi}_p(k)_i + \beta(k)\left[\pi_p(k)_i - \overline{\pi}_p(k)_i\right]$;
    if $\sum \pi_p(k)_i \hat{q}_k(s,a) > \sum \overline{\pi}_p(k)_i \hat{q}_k(s,a)$ **then**
      $\delta = \delta_w$;
    else
      $\delta = \delta_l$;
    end if
  end for
  $\delta_p = \delta_l$, $\delta_m = -1 - \delta_p$;
  if $a_p := \arg\max_i \pi_p(k)$ **then**
    $\pi_p(k)_i = \min\left(1, \pi_p(k)_i + \delta_p\right)$;
  else
    $\pi_p(k)_i = \max\left(0, \pi_p(k)_i + \delta_m\right)$;
  end if
  update $\overline{\pi}_e(k)$ and $\pi_e(k)$ the same as the above steps;
  $a_p(k) = a_{p_{i^*}} = \arg\max_i \pi_p(k)$;
  $a_e(k) = a_{e_{j^*}} = \arg\max_j \pi_e(k)$;
  $k = k + 1$;
end while

Simulations

The remote environmental monitoring units (REMUS) AUVs have been developed for almost 30 years, which hold the lead in the AUV researches [42]. Among various REMUS AUVs, the REMUS 600 AUV stands out. It is a reliable and cost-effective platform. For example, the REMUS 600 AUV can travel to the ranges from 150 to 200 km at a speed of about 2.0 m/s. It can also move at a maximum depth of 600 m. Therefore, the REMUS 600 AUV is utilized to define the parameters for the pursuit AUVs and the invading AUV in the proposed model. The relevant parameters assumed for these vehicles, along with the key algorithmic parameters, are presented in Table 3.

Proper deployment ensures that the AUVs can perform their task. Hydroid Company and WHOI have developed a slideway deployment and retrieval device that incorporates low-cost lightweight composite rope technology. This device effectively addresses the autonomous deployment challenges of REMUS-600 AUVs, and creates a towed rope biting AUV autonomous deployment system [43]. Based on this deployment method, Fig. 3 illustrates the deployment for the pursuit AUVs in the proposed model. As shown in Table 4, three loosely-distributed systems are assumed accordingly. The protected area is assumed to have dimensions of 40m × 40m × 40m.

Simulations under different RL algorithms

Convergence ability is crucial for enabling real-time decision-making. To evaluate this, the WoLF-PHC algorithm is compared with three other relevant RL algorithms: Minimax-Q, Advantage actor critic (A2C), and Proximal policy optimization (PPO).

First of all, analyzing the computational complexity of each algorithm helps us understanding how efficiently they can converge. This is helpful for evaluating their suitability for real-time decision-making. The time complexity per update for WoLF-PHC is $O(S \times A)$, where S and A represent the number of states and actions, respectively. This is similar to the complexity of Minimax-Q algorithm. However, while the WoLF mechanism causes minor computational overhead by tracking separate learning rates for updates, it enhances the responsiveness of the WoLF-PHC in dynamic environments. This results in slightly longer convergence times compared to Minimax-Q, but the trade-off improves decision-making

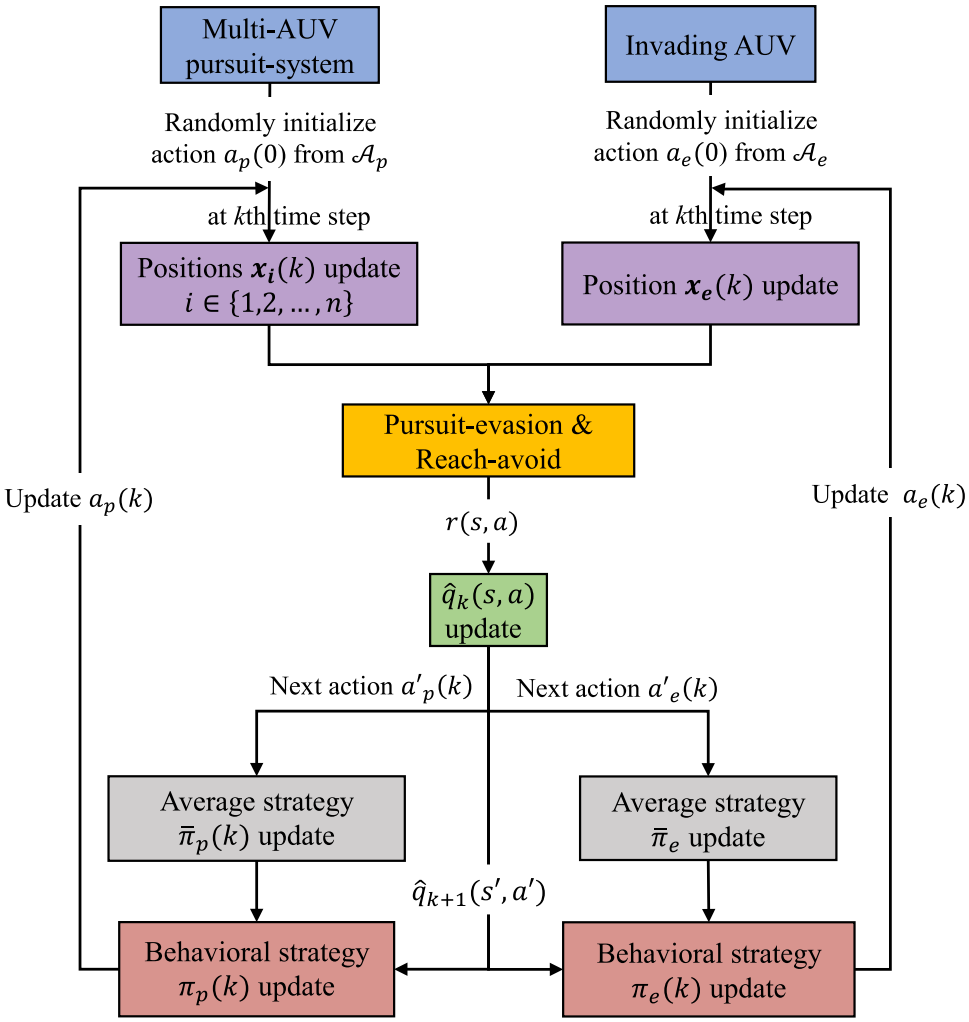


Fig. 2 Learning process of the WoLF-PHC in the proposed ZSSG-based model

accuracy. As A2C and PPO rely on neural networks for policy and value function estimation, their time complexity depends on the size of the neural network. This leads to a time complexity of $O(S \times A \times N)$, where N denotes the number of network layers and parameters. Consequently, in environments with large state spaces, the computational overhead of A2C and PPO is significantly higher than that of WoLF-PHC and Minimax-Q.

In terms of space complexity, both WoLF-PHC and Minimax-Q share a space complexity of $O(S \times A)$. However, WoLF-PHC incurs a slight increase in memory requirements due to the tracking of dual learning rates, which leads to more nuanced policy updates. A2C and PPO require additional storage for neural network parameters. This results in a space complexity of $O(N)$, which is much larger, especially in complex tasks. Overall, WoLF-PHC strikes a balance between computational efficiency and enhanced decision-making capabilities.

With the above understanding of the computational requirements for the four algorithms, we now turn to the simulation setup for the convergence analysis. The initial position of the invading AUV is set as $\boldsymbol{x_e}(0) := (5, 40, 40)$,

Table 3 Parameter settings for vehicles and algorithm

Parameter	Definition	Value
v_p	Scalar velocity of the pursuit AUV	2.0 m/s
v_e	Scalar velocity for the invading AUV	1.6 m/s
l	Length of the pursuit AUV	4.3 m
d_s	Safe distance between the two AUVs	10.32 m
d_c	Captured distance for the pursuit AUV	12 m
d_a	Attacked distance for the invading AUV	6 m
β	Learning rate	0.01
γ	Discount factor	0.9
δ_w	Learning rate (win)	0.0025
δ_l	Learning rate (lose)	0.01

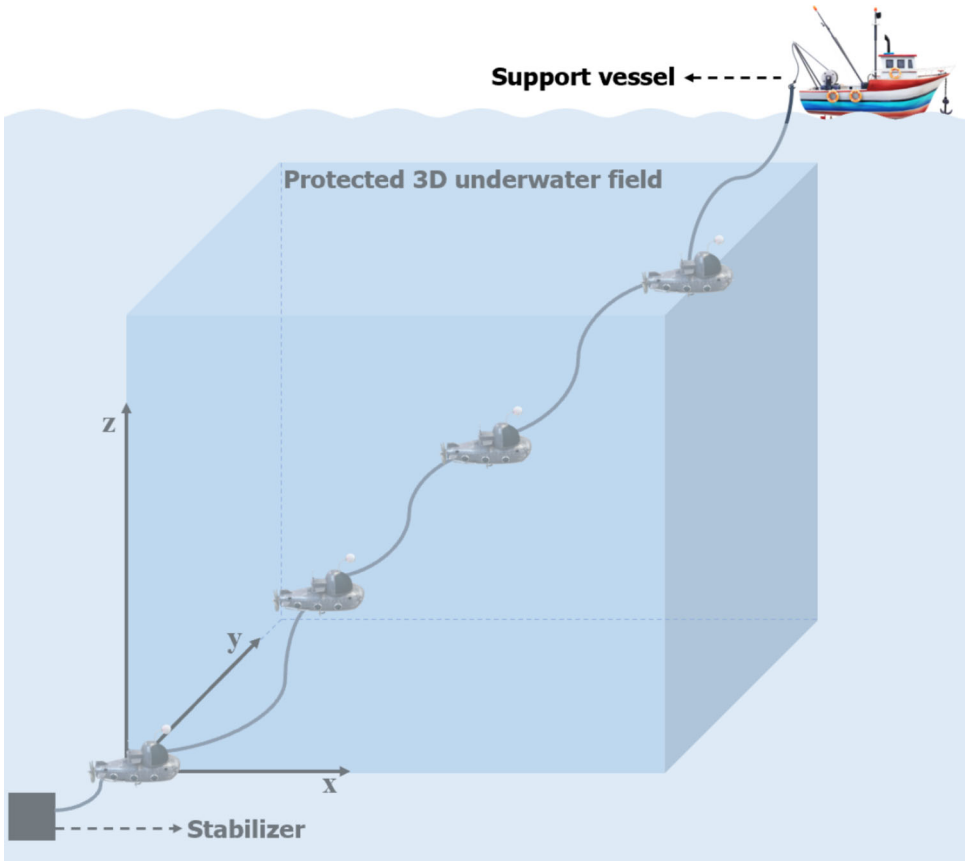


Fig. 3 Deployment of pursuit AUVs within the protected area

and its point of attack is defined as $\boldsymbol{x_a} := (35, 35, 35)$. Therefore, at $k = 0$, in the introduced loosely distributed systems, the reward values are shown in Fig. 4.

Form Fig. 4, it is evident that the WoLF-PHC algorithm demonstrates the best convergence performance, which can be attributed to its WoLF scheme. Besides, with calculation being more complex in the A2C and PPO algorithms, more steps would be cost to reach the equilibrium point. It can be concluded that simpler RL approaches may be more efficient for the simple tasks and closer to the real-time implementation. Therefore, compared with other state-of-the-art RL algorithms, the WoLF-PHC algorithm takes the advantage in the studied target pursuit model for the multi-AUV system. To enhance the computational efficiency of WoLF-PHC in complex scenarios, optimizations like parallelizing policy updates and applying state-space reduction techniques can be explored.

To further highlight the advantages of WoLF-PHC, a comparative analysis was conducted with two other commonly used algorithms: Minimax Q-learning and the classical linear programming solution. The comparison included both qualitative and quantitative evaluations. From a qualitative perspective, WoLF-PHC stands out due to its ability to adjust its strategy dynamically. It achieves the dynamic adjustment by using a learning rate that evolves over time. In contrast, while minimax Q-learning can learn from experience, it is unable to inherently adjust the learning rate in response to dynamic environments. This drawback can limit its performance in uncertain settings. The classical linear programming solution, though optimal in static or predefined scenarios, lacks the ability to adapt to environmental changes. As a result, WoLF-PHC offers greater flexibility and robustness. This makes it better suited for dynamic tasks that require continuous learning and adaptation.

On the quantitative side, the comparison focused on two key indicators: computational efficiency and success rate of capture. Computational efficiency, measured by average time per simulation, is critical for multi-agent systems, especially in real-time decision-making. Higher efficiency means tasks are completed faster. Success rate of capture measures task success. Capturing the target is the main goal for multi-agent target pursuit system. Under conditions where $\boldsymbol{x_e}(0)$ and $\boldsymbol{x_a}$ are randomly chosen within the protected area, 1000-time simulations are conducted under the three methods, respectively. Simulation results about the computational efficiency and the success rate of capture are presented in Fig. 5.

As shown in Fig. 5, it is obvious that as the number of pursuit AUVs increases, the average time per simulation under the WoLF-PHC remains the lowest. Compared with the linear programming algorithm and the minimax Q-learning

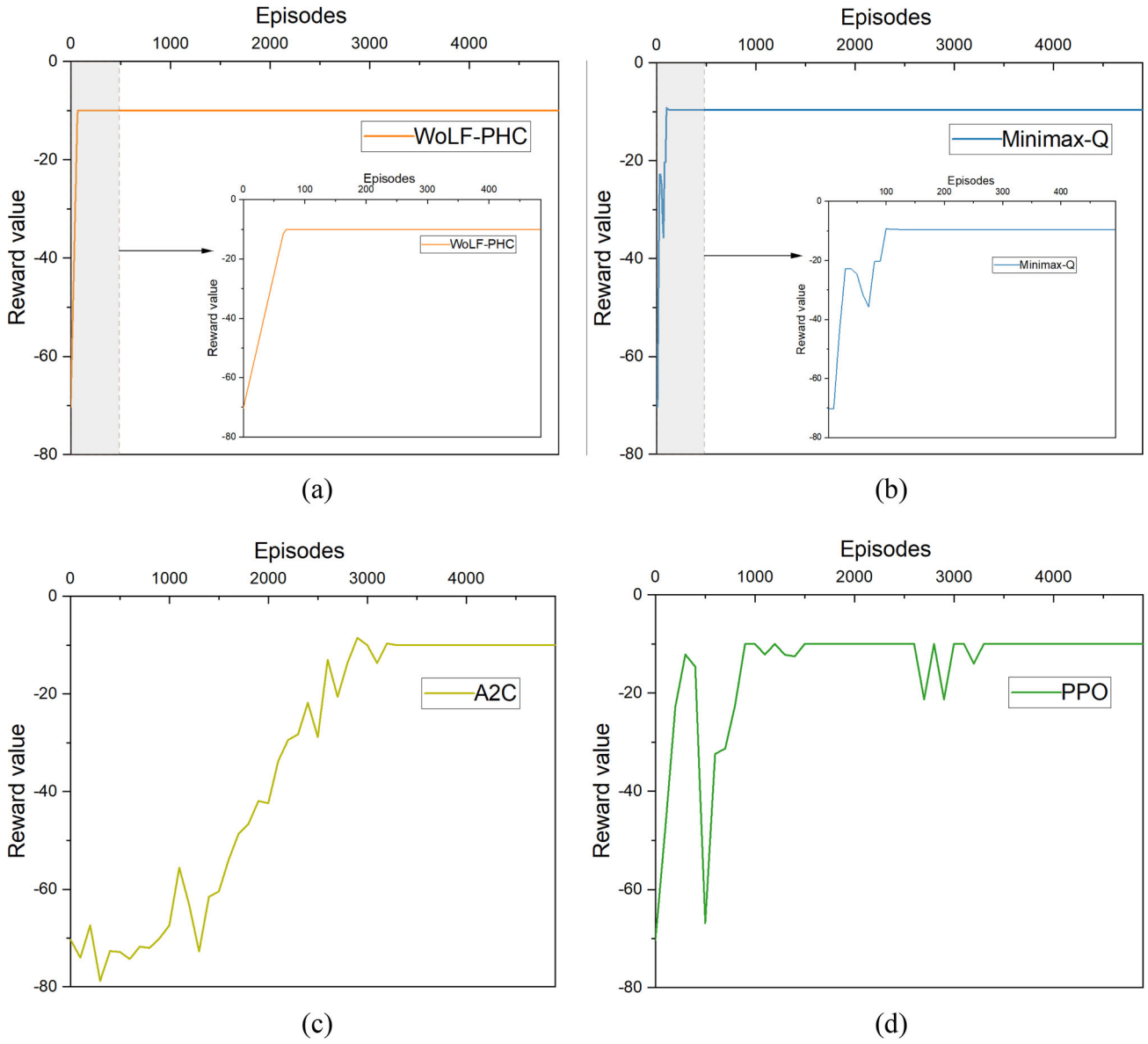


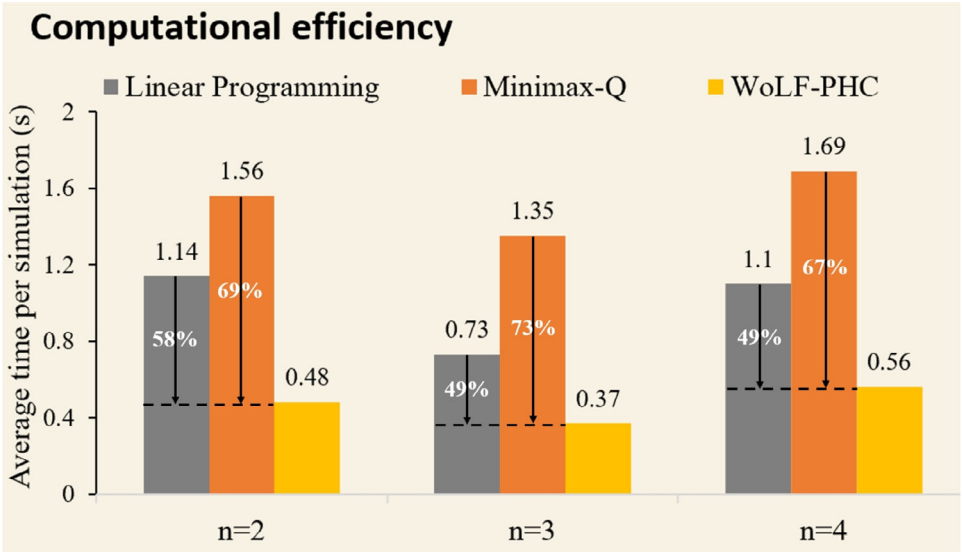
Fig. 4 Convergence analysis for the target pursuit reward under the four different RL algorithm: **a** WoLF-PHC, **b** Minimax-Q, **c** A2C, and **d** PPO

method, the WoLF-PHC significantly improves calculation efficiency, achieving a reduction of approximate 50% and 70%, respectively. Besides, according to the results shown in Fig. 5b, there is no obvious difference between the WoLF-PHC algorithm and the minimax Q-learning in terms of the success rate of capture, while the classic solution lags behind. In summary, the results highlight that WoLF-PHC not only excels in computational efficiency but also maintains strong performance in terms of capture success rate.

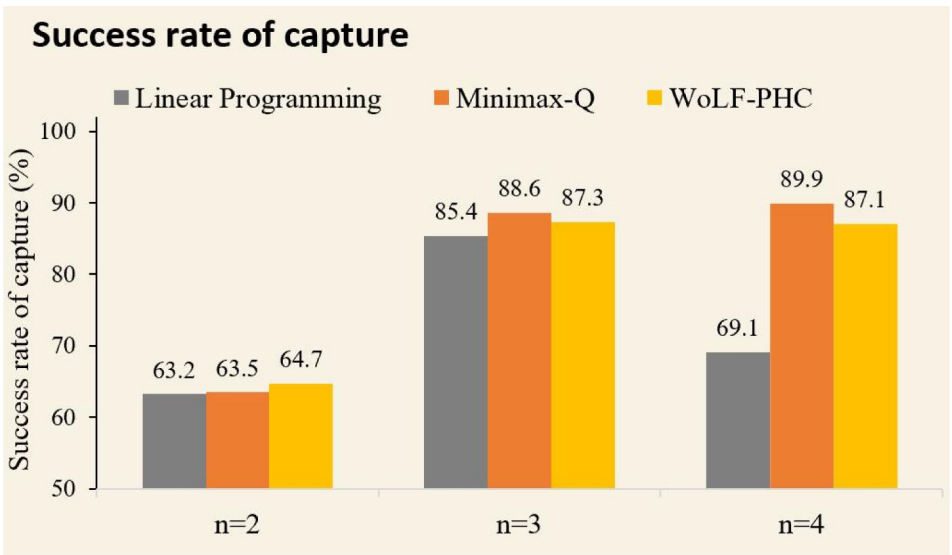
Simulations under two distribution types

As the deployment for the pursuit AUVs is not the focus of this paper, only two distribution type, the loosely-distributed type and tightly-distributed type, are evaluated. The positions have been provided in Table 4 for the loosely-distributed system. Positions in Table 5 are set for the tightly-distributed system. 1000 times simulations are conducted for each distribution type under the proposed approach.

As shown in Table 6, two valuable findings could be obtained. First, the tightly-distributed system outperforms the loosely-distributed system in both the success rate of capture and the computational efficiency. This observation aligns with the real-world scenario, as a tightly-distributed system implies that data or elements are clustered more closely together. The clustered data/elements result in smaller variations or deviations from the average. Second, the success rate of capture does not show significant improvement with an



(a)



(b)

Fig. 5 Pursuit performance under the three algorithms in the loosely distributed system in terms of the **a** computational efficiency, and **b** success rate of capture

Table 4 Deployment in the loosely distributed system

Loosely-distributed system	Initial positions for the pursuit AUVs			
	$\boldsymbol{x}_1(0)$	$\boldsymbol{x}_2(0)$	$\boldsymbol{x}_3(0)$	$\boldsymbol{x}_4(0)$
$n = 2$	(0,0,0)	(40,40,40)	N/A	N/A
$n = 3$	(0,0,0)	(20,20,20)	(40,40,40)	N/A
$n = 4$	(0,0,0)	(20,20,20)	(30,30,30)	(40,40,40)

Table 5 Initial positions for the pursuit AUVs in the tightly distributed system

Tightly-distributed system	Initial positions for the pursuit AUVs			
	$\boldsymbol{x}_1(0)$	$\boldsymbol{x}_2(0)$	$\boldsymbol{x}_3(0)$	$\boldsymbol{x}_4(0)$
$n = 2$	(20,20,20)	(25,25,25)	N/A	N/A
$n = 3$	(15,15,15)	(20,20,20)	(25,25,25)	N/A
$n = 4$	(15,15,15)	(20,20,20)	(25,25,25)	(30,30,30)

Table 6 Comparison of the two types distributed pursuit system

Distribution type	$n = 2$		$n = 3$		$n = 4$	
	$r_s(\%)$*	$t_a(s)$**	$r_s(\%)$	$t_a(s)$	$r_s(\%)$	$t_a(s)$
Loosely-distributed	64.7	0.48	87.3	0.37	87.1	0.56
Tightly-distributed	93.1	0.18	92.5	0.25	90.7	0.47

*r_s: success rate of capture
**t_a: average time per simulation

increasing number of AUVs. This indicates that the method performs well in scenarios with fewer AUVs. The reduced scalability in larger system is likely due to current limitations in coordination mechanism. To address this limitation, improvements in communication protocols are expected.

Robustness under different evasion action-sets

To test the robustness of the proposed approach under the different action-sets of the invading AUV, two newly action-sets are designed for the invading AUV in the tightly-distributed system with $n = 3$. These two new actions-sets are named by the fixed evasion action-set and the enriched evasion action-set. In the fixed action-set, the invading vehicle would move to its target all the time. The enriched action-set is more complex than the original set. The enriched motions are designed as follows and shown in Fig. 6.

$\boldsymbol{u}_{\boldsymbol{e}_1}$: the invading AUV evades from the nearest pursuit AUV.

$\boldsymbol{u}_{\boldsymbol{e}_2}$: the invading AUV adopts the collective evasion-motion, which is explained in Definition 5.

$\boldsymbol{u}_{\boldsymbol{e}_3}$: the invading AUV heads directly toward its target.

$\boldsymbol{u}_{\boldsymbol{e}_4}$: the direction of the invading AUV is the angle bisector formed by $\boldsymbol{u}_{\boldsymbol{e}_1}$ and $\boldsymbol{u}_{\boldsymbol{e}_3}$.

Definition 5 (*Collective Evasion-Motion*) From the view of the invading vehicle, all angles formed between the two adjacent pursuit AUVs are taken into account in the collective evasion-motion. Besides, the moving direction of the invading vehicle is related to the parallelogram of the maximum angle formed by the two adjacent pursuit AUVs. The purpose of this motion is to enable the invading vehicle to move immediately away from the entire pursuit team, rather than just from one pursuit AUV. The calculation for this kind of

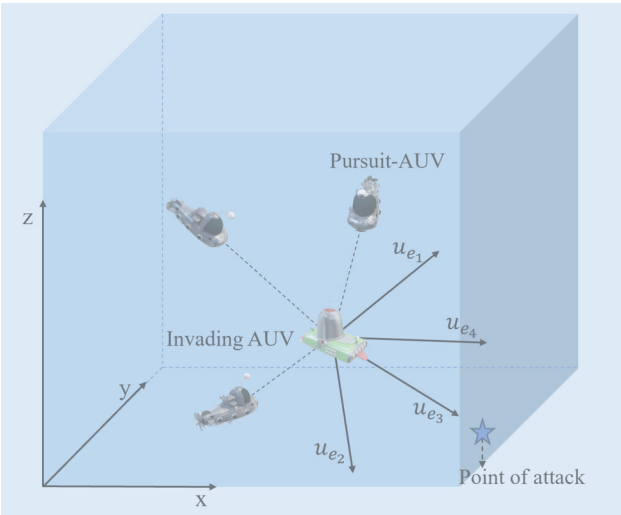


Fig. 6 Enriched evasion-motions for the invading AUV

evasion-motion is as follows: $\phi_i := \angle(\boldsymbol{x_e}, \boldsymbol{x_i})$ is set to represent the angle of the vector $(\boldsymbol{x_e} - \boldsymbol{x_i})$. $\theta_i := \phi_{i+1} - \phi_i$ denotes the angle between two adjacent pursuit AUVs. If $i = n$, ϕ_{i+1} is equal to phi_1. Therefore, the angle for the $\boldsymbol{u}_{\boldsymbol{e}_2}$ could be obtained by Eq. (16).

$$\angle \boldsymbol{u}_{\boldsymbol{e}_2} = \theta_{i_{mh}} + \theta_{i_m}, \tag{16}$$

where i_m is the index when θ_i is taken to its maximum value. $\theta_{i_{mh}}$ is the half of the θ_{im}.

Based on the above introduction, under the varying pursuit velocities, simulations are conducted under the three designed evasion action-sets. The results are shown in Fig. 7, where we can find that there is no obvious difference between the original action-set $\mathcal{A}_e$ and the fixed $\mathcal{A}_e$. However, the success rate in the enriched $\mathcal{A}_e$ is the highest among the three

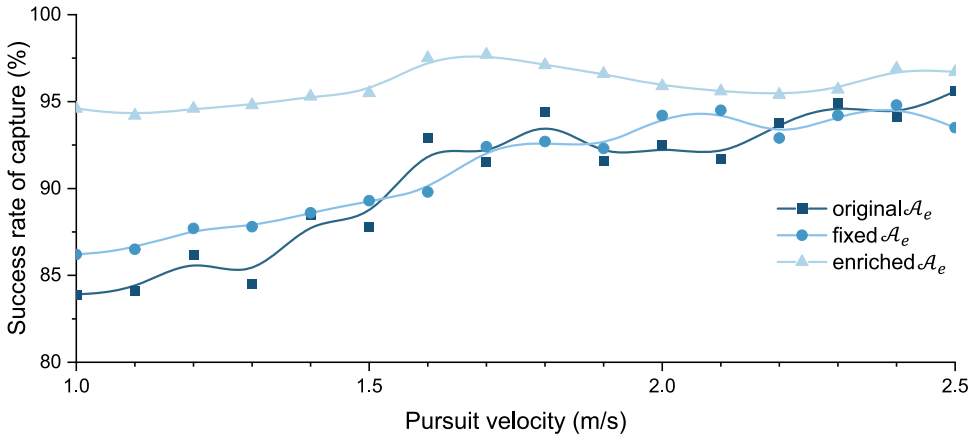


Fig. 7 Success rate of capture under three evasion action-sets: original $\mathcal{A}_e$, fixed $\mathcal{A}_e$, and enriched $\mathcal{A}_e$

sets. Besides, the change in the pursuit velocity has little impact on the success rate under the enriched $\mathcal{A}_e$. These conclusions validate the robustness of the proposed approach. They demonstrate that the proposed pursuit model can adapt to different evasion actions.

Simultaneous analysis and potential implementations

To achieve simultaneous coordination among the pursuit AUVs, two pursuit mechanisms are introduced, named by the leader–follower pursuit and the attack-protect pursuit.

Definition 6 (*Leader–Follower Pursuit*) The active AUV chosen by the ZSSG-based system model would be treated as the leader during the pursuit. The other AUVs would follow its motion. If the leader-AUV has been determined by the index i^*, the motion $\boldsymbol{u_i}$ for the ith pursuit AUV in the leader–follower framework could be obtained as follows.

$$\boldsymbol{u_i} = \begin{cases} \frac{\boldsymbol{x_e} - \boldsymbol{x_i}}{\|\boldsymbol{x_e} - \boldsymbol{x_i}\|}, \, if \, i = i^* \\ \boldsymbol{u}_{i^*}, \, others \end{cases} \tag{17}$$

Definition 6 (*Attack-Protect Pursuit*) The same as the relay-pursuit, only one AUV would be selected to pursuit the invading AUV. However, to protect the point of attack within the protected area, the other AUVs would move to the target. Therefore, the motion $\boldsymbol{u_i}$ for the ith pursuit AUV in the attack-protect mechanism could be obtained as follows:

$$\boldsymbol{u_i} = \begin{cases} \frac{\boldsymbol{x_e} - \boldsymbol{x_i}}{\|\boldsymbol{x_e} - \boldsymbol{x_i}\|}, \, if \, i = i^* \\ \frac{\boldsymbol{x_a} - \boldsymbol{x_i}}{\|\boldsymbol{x_a} - \boldsymbol{x_i}\|}, \, others \end{cases}. \tag{18}$$

According to the above statement and definitions, performance under the three pursuit modes is shown in Table 7. Parameter setting is the same as that in the simulations about the robustness under the different evasion action-sets. In contrast to the performance evaluation in previous experiments, we introduce the indicator r_a. This indicator denotes rate of the invading AUV's attack-point being attacked. It evaluates the ability of the pursuit system to safeguard unknown points from being attacked by an invading AUV. This new evaluation indicator provides a detailed and in-depth analysis of the pursuit performance in different pursuit modes.

From the simulation results shown in Table 7, it is obvious that the simultaneous coordination-based pursuit modes perform better than the pure relay-pursuit. Although the leader–follower pursuit has the biggest success rate of capture, the attack-protect pursuit possesses the smallest rate of target being attacked and achieves a relatively good success rate of capture. Therefore, under different scenarios and different aims, the most suitable simultaneous coordination methods would be different. Besides, these conclusions also clarify that the proposed pursuit model could adapt to different simultaneous pursuit modes and achieve a good pursuit performance.

The above results demonstrate that our proposed method achieves a high success rate of capture. This validates its effectiveness in multi-AUV target pursuit scenarios. Additionally, the method consistently delivers strong performance across diverse evasion action-sets and different pursuit modes. The high success rate shows that the system effectively pursues and captures targets. This capability is also crucial for underwater tasks like marine surveillance, and search-and-rescue missions. Moreover, the proposed method also exhibits high computational efficiency. This makes the method feasible for implementation and suitable for deployment in resource-constrained environments, especially for the AUVs with limited processing power and battery life.

In summary, the results confirm that our proposed method is both effective and efficient. It achieves a high success rate of capture, and also demonstrates strong computational efficiency. These factors make it highly feasible for practical use in multi-AUV target pursuit.

Table 7 Performance under three pursuit modes

A_p	$n=2$		$n=3$		$n=4$	
	$r_s(\%)$	$r_a(\%)$*	$r_s(\%)$	$r_a(\%)$	$r_s(\%)$	$r_a(\%)$
Relay-pursuit	93.1	5.9	92.5	5.5	90.7	9.9
Leader–follower	94.4	4.4	95.8	3.5	92.7	6.3
Attack-protect	95.0	3.6	93.8	3.2	93.3	3.5

*r_a: rate of the invading AUV's attack-point being attacked

Conclusions

This paper has introduced a novel approach to determine the optimal pursuit motions for the multi-AUV system. At first, a two-player ZSSG-based framework is employed to construct the pursuit system model for a multi-AUV system. Then, to relax the dependency of the model on the parameter setting, the WoLF-PHC algorithm is introduced and applied to the ZSSG-based pursuit model. Simulation results validate the efficacy of the proposed method in providing the optimal pursuit motions. Compared with the other RL algorithms, the WoLF-PHC algorithm possesses the quickest convergence speed in obtaining the optimal reward value. Furthermore, the WoLF-PHC approach outperforms the two commonly used algorithms in terms of the computational efficiency without the cost of the success rate of capture. At last, the impact of multi-AUV distribution on the proposed method is assessed. The robustness and the simultaneity of the proposed pursuit system model are also validated and explored.

In conclusion, a promising approach has been proposed for optimizing decision-making for multi-AUV to conduct target pursuit in an uncertain 3D environment. The proposed approach has potential applications in diverse domains, such as underwater surveillance, search and rescue operations, and environmental monitoring. Nonetheless, the simulation results reveal that as the number of AUVs increases, the capture success rate does not necessarily improve, which contradicts real-world expectations. To ensure better scalability and improved performance in larger-scale systems, future efforts will focus on optimizing coordination mechanisms and enhancing communication protocols among AUVs. Additionally, since this work is based on simulations, future research would also include validating the method in the real-world settings.

Acknowledgements This work was supported by Zhejiang Key R&D Program no. 2021C03157, start-up funding from Westlake University under grant number 041030150118 and Scientific Research Funding Project of Westlake University under Grant no. 2021WUFP017. The authors would like to thank the anonymous reviewers for their valuable insights and feedback.

Data availability The data used in this study is available upon reasonable request.

References

1. Li G, Chen X, Zhou F, Liang Y, Xiao Y, Cao X et al (2021) Self-powered soft robot in the Mariana Trench. Nature 591:66–71. https://doi.org/10.1038/s41586-020-03153-z
2. Wu G, Xu T, Sun Y, Zhang J (2022) Review of multiple unmanned surface vessels collaborative search and hunting based on swarm intelligence. Int J Adv Robot Syst 19:1–20. https://doi.org/10.1177/17298806221091885
3. Wang Y, Zhong F, Xu J, Wang Y (2022) ToM2C: target-oriented multi-agent communication and cooperation with theory of mind. In: Int Conf Learn Represent (ICLR). https://doi.org/10.48550/arXiv.2111.09189
4. Zhang Q, Song X, Song S, Stojanovic V (2023) Finite-Time sliding mode control for singularly perturbed PDE systems. J Franklin I 360:841–861. https://doi.org/10.1016/j.jfranklin.2022.11.037
5. Tao Y, Tao H, Zhuang Z, Stojanovic V, Paszke W (2024) Quantized iterative learning control of communication-constrained systems with encoding and decoding mechanism. Trans I Meas Control 46:1943–1954. https://doi.org/10.1177/01423312231225782
6. Peng Z, Song X, Song S, Stojanovic V (2023) Hysteresis quantified control for switched reaction–diffusion systems and its application. Complex Intell Syst 9:7451–7460. https://doi.org/10.1007/s40747-023-01135-y
7. Lin W, Qu Z, Simaan MA (2015) Nash strategies for pursuit-evasion differential games involving limited observations. IEEE Trans Aero Elec Sys 51:1347–1356. https://doi.org/10.1109/TAES.2014.130569
8. Zheng Z, Zhang P, Yuan J (2023) Nonzero-sum pursuit-evasion game control for spacecraft systems: a Q-learning method. IEEE Trans Aero Elec Sys 59:3971–3981. https://doi.org/10.1109/TAES.2023.3235873
9. Talebi S, Simaan MA (2017) Multi-pursuer pursuit-evasion games under parameters uncertainty: a Monte Carlo approach. In: 2017

12th System of systems engineering conference (SoSE). IEEE, pp 1–6. https://doi.org/10.1109/SYSOSE.2017.7994937
10. Deng X, Li N, Mguni D, Wang J, Yang Y (2022) On the complexity of computing Markov perfect equilibrium in general-sum stochastic games. Natl Sci Rev. https://doi.org/10.1093/nsr/nwac256
11. Shi H, Zhai L, Wu H, Hwang M, Hwang KS, Hsu HP (2020) A multitier reinforcement learning model for a cooperative multiagent system. IEEE Trans Cogn Dev Syst 12:636–644. https://doi.org/10.1109/TCDS.2020.2970487
12. Shi D, Sauter MZ, Kralik JD (2009) Distributed, heterogeneous, multi-agent social coordination via reinforcement learning. In: 2009 IEEE international conference on robotics and biomimetics (ROBIO). IEEE, pp 653–58. https://doi.org/10.1109/ROBIO.2009.5420595
13. Bowling M, Veloso M (2001) Convergence of gradient dynamics with a variable learning rate. In: Proceedings of the eighteenth international conference on machine learning (ICML), pp 27–34 https://doi.org/10.5555/645530.655659
14. Xi L, Yu T, Yang B, Zhang X (2015) A novel multi-agent decentralized win or learn fast policy hill-climbing with eligibility trace algorithm for smart generation control of interconnected complex power grids. Energ Convers Manage 103:82–93. https://doi.org/10.1016/j.enconman.2015.06.030
15. Mai T, Yao H, Zhang N, Xu L, Guizani M, Guo S (2021) Cloud mining pool aided blockchain-enabled Internet of Things: an evolutionary game approach. IEEE Trans Cloud Comput 11:692–703. https://doi.org/10.1109/TCC.2021.3110965
16. Wishart D (1966) Differential games. A mathematical theory with applications to warfare and pursuit, control and optimization. Phys Bull 17:60. https://doi.org/10.1088/0031-9112/17/2/009
17. Selvakumar J, Bakolas E (2022) Min-Max Q-learning for multiplayer pursuit-evasion games. Neurocomputing 475:1–14
18. Zhang R, Zong Q, Zhang X, Dou L, Tian B (2022) Game of drones: Multi-UAV pursuit-evasion game with online motion planning by deep reinforcement learning. IEEE Trans Neur Net Lear 34:7900–7909. https://doi.org/10.1109/TNNLS.2022.3146976
19. Zhang Y, Guizani M (2011) Game theory for wireless communications and networking. CRC Press, Boca Raton. https://doi.org/10.5555/1942844
20. Selvakumar J, Bakolas E (2019) Feedback strategies for a reach-avoid game with a single evader and multiple pursuers. IEEE Trans Cybern 51:696–707. https://doi.org/10.1109/TCYB.2019.2914869
21. Wang Y, Dong L, Sun C (2020) Cooperative control for multiplayer pursuit-evasion games with reinforcement learning. Neurocomputing 412:101–114. https://doi.org/10.1016/j.neucom.2020.06.031
22. Vrancx P, Verbeeck K, Nowé A (2008) Decentralized learning in Markov games. IEEE Trans Syst Man Cy B 38:976–981. https://doi.org/10.1109/TSMCB.2008.920998
23. Sastry PS, Phansalkar VV, Thathachar M (1994) Decentralized learning of Nash equilibria in multi-person stochastic games with incomplete information. IEEE Trans Syst Man Cy 24:769–777. https://doi.org/10.1109/21.293490
24. Du W, Guo T, Chen J, Li B, Zhu G, Cao X (2021) Cooperative pursuit of unauthorized UAVs in urban airspace via Multi-agent reinforcement learning. Transp Res C-Emer. https://doi.org/10.1016/J.TRC.2021.103122
25. Pérolat J, Piot B, Geist M, Scherrer B, Pietquin O (2016) Softened approximate policy iteration for Markov games. In: Int Conf Mach Learn (ICML). PMLR, pp 1860–8 https://doi.org/10.5555/3045390.3045587
26. Sidford A, Wang M, Yang L, Ye Y (2020) Solving discounted stochastic two-player games with near-optimal time and sample complexity. In: Int Conf Artif Intell Stat (AISTATS). PMLR, pp 2992–3002. https://doi.org/10.48550/arXiv.1908.11071
27. Littman ML (1994) Markov games as a framework for multi-agent reinforcement learning. In: Mach Learn Proc. Elsevier, pp 157–63 https://doi.org/10.1016/B978-1-55860-335-6.50027-1
28. Hong L, Cui W (2023) Strategy determination for multiple USVs: a min-max Q-learning approach. In: Int Conf Neur Comput Adv Appl Springer, pp 403–17. https://doi.org/10.1007/978-981-99-5847-4_29
29. Fan J, Wang Z, Xie Y, Yang Z (2020) A theoretical analysis of deep Q-learning. In: Learn Dynam Control (L4DC). PMLR, pp 486–489. https://doi.org/10.48550/arXiv.1901.00137
30. Bowling M, Veloso M (2001) Rational and convergent learning in stochastic games. In: Int Jt Conf Artif Iintell (IJCAI). Citeseer, pp 1021–1026. https://doi.org/10.5555/1642194.1642231
31. Conitzer V, Sandholm T (2007) AWESOME: a general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. Mach Learn 67:23–43. https://doi.org/10.1007/s10994-006-0143-1
32. Nguyen DT, Kumar A, Lau HC (2017) Policy gradient with value function approximation for collective multiagent planning. Adv Neur Inform Process Syst (NIPS). https://doi.org/10.48550/arXiv.1804.02884
33. Sun Z, Sun H, Li P, Zou J (2023) Cooperative strategy for pursuit-evasion problem with collision avoidance. Ocean Eng. https://doi.org/10.1016/j.oceaneng.2022.112742
34. Qu X, Gan W, Song D, Zhou L (2023) Pursuit-evasion game strategy of USV based on deep reinforcement learning in complex multi-obstacle environment. Ocean Eng. https://doi.org/10.1016/j.oceaneng.2023.114016
35. Hong L, Cui W, Chen H, Song C, Li W (2024) Maneuver planning for multiple pursuit intelligent surface vehicles in a sequence of zero-sum pursuit-evasion games. J Mar Sci Eng 12:1–21. https://doi.org/10.3390/jmse12071221
36. Busoniu L, Babuska R, De Schutter B (2008) A comprehensive survey of multiagent reinforcement learning. IEEE Trans Syst Man Cy C 38:156–172. https://doi.org/10.1109/TSMCC.2007.913919
37. Zhang K, Kakade S, Basar T, Yang L (2020) Model-based multiagent rl in zero-sum Markov games with near-optimal sample complexity. Adv Neur Inform Process Syst (NIPS) 33:1166–1178. https://doi.org/10.48550/arXiv.2007.07461
38. Fink AM (1964) Equilibrium in a stochastic n-person game. J Sci Hiroshima Univ Ser A-I Math 28:89–93. https://doi.org/10.32917/hmj/1206139508
39. Gao X, Chan KW, Xia S, Zhang X, Zhang K, Zhou J (2021) A multiagent competitive bidding strategy in a pool-based electricity market with price-maker participants of WPPs and EV aggregators. IEEE Trans Ind Inform 17:7256–7268. https://doi.org/10.1109/TII.2021.3055817
40. Sayin M, Zhang K, Leslie D, Basar T, Ozdaglar A (2021) Decentralized Q-learning in zero-sum Markov games. Adv Neur Inform Process Syst (NIPS) 34:18320–18334. https://doi.org/10.48550/arXiv.2106.02748
41. Bloembergen D, Tuyls K, Hennes D, Kaisers M (2015) Evolutionary dynamics of multi-agent learning: a survey. J Artif Intell Res 53:659–697. https://doi.org/10.1613/jair.4818
42. Fiester C, Gomez-Ibanez D, Grund M, Purcell M, Jaffre F, Forrester N et al (2019) A modular, compact, and efficient next generation remus 600 auv. In: OCEANS 2019-Marseille. IEEE, pp 1–6 https://doi.org/10.1109/OCEANSE.2019.8867248
43. Zheng R, Xin C, Tang Z, Song T (2020) Review on the platform technology of autonomous deployment of AUV by USV. Acta Aliment Hung. https://doi.org/10.3969/j.issn.1000-1093.2020.08.022