Applied Ocean Research 122 (2022) 103106

Contents lists available at ScienceDirect

Applied Ocean Research

journal homepage: www.elsevier.com/locate/apor

Coordination of distributed unmanned surface vehicles via model-based reinforcement learning methods

Runlong Miao [a], Lingxiao Wang [b], Shuo Pang [b,*]

[a] National Key Laboratory of Autonomous Underwater Vehicle, College of Shipbuilding Engineering, Harbin Engineering University, Harbin 15001, China
[b] Department of Electrical Engineering and Computer Science, Embry-Riddle Aeronautical University, Daytona Beach, FL, 32124, USA

ARTICLE INFO

Keywords:
Multi-agent coordination
Unmanned surface vehicles
Reinforcement learning
Path planning

ABSTRACT

This article presents a coordination algorithm for organizing a fleet of unmanned surface vehicles (USVs) to search multiple moving object targets in the ocean environment. During the fleet maneuver, USVs can exchange local sensing information through a wireless communication network. Based on both received and self-perceived information, a USV constructs a grid confidence map, which reflects how well the USV fleet perceives on every region of the search area. Then, the USV coordination is modeled as a reinforcement learning (RL) problem, where reward functions are defined based on the information obtained from the grid confidence map. Therefore, USVs are encouraged to explore new regions and prevented visiting the already searched areas. Search routes of USVs are calculated via a policy-iteration based path planning algorithm, while inter-vehicle collisions are avoided by applying policy constraints. Real-world experiments were conducted in the ocean environment to evaluate the validity of the proposed method. Compared to the conventional formation control strategy and the uncoordinated algorithm, experiment results show that the proposed method is more intelligent and efficient for searching object targets.

1. Introduction

Unmanned surface vehicles (USVs) are marine vessels developed to support maritime missions. Characterized by low cost, high mobility, and a high degree of autonomy, USVs have been implemented in various maritime tasks, such as search and rescue, marine surveys, environmental monitoring, and offshore installation protection (Caccia et al., 2005; Roberts and Sutton, 2006; Naeem et al., 2008). The capability of a single USV is constrained due to the limited payload capacity and short endurance times. For the large-scale and complex ocean missions, deploying a fleet of USVs is more applicable than a single USV, considering the wider mission area, improved system robustness, and increased mission efficiencies (Liu and Bucknall, 2016).

Despite the practical potentials and advantages, deploying a fleet of USVs instead of one raises new challenges and problems. Considerations for designing a multi-USV system, such as the requirement of avoiding inter-vehicle collisions, the scarcity of communication bandwidth in ocean environments, and the demand for coordinating behaviors of USVs to achieve optimal efficiency, are necessary to accommodate. The central to address these problems is an effective coordination algorithm, which organizes USVs to perform planned missions while avoiding collisions and meeting communication constraints.

Inspired by formation behaviors of animals, such as flocking of birds and swarming of ants, formation control algorithms (Chen and Wang, 2005) have been widely developed to coordinate multi-agent systems. In this type of algorithms, agents are controlled to perform operations collaboratively while maintaining the desired formation. In the field of USV coordination, typical implementations of formation control strategies include the path following Yin et al. (2016) and target tracking (Yang et al., 2014), where USVs are driven to follow predefined paths or track targets while holding the desired formation pattern. Various control strategies have been proposed to maintain the USV formation pattern and reject environmental disturbances, including classic control theories, e.g., sliding mode control (Li et al., 2018), adaptive control (Almeida et al., 2010), output feedback control (Peng et al., 2015), and constrained control (Peng et al., 2017); artificial intelligence theories, e.g., neural networks (Peng et al., 2012) and deep reinforcement learning methods (Meyer et al., 2020); fuzzy theories, e.g., Takagi–Sugeno (T–S) fuzzy controllers (Wang et al., 2018; Peng et al., 2020).

In this work, our concentration is to design a coordination algorithm that organizes a fleet of USVs to search multiple object targets in the ocean environment. Potential applications of the proposed algorithm include marine surveys or rescue tasks, where object targets

* Corresponding author.
E-mail addresses: lingxiaw@my.erau.edu (L. Wang), shuo.pang@erau.edu (S. Pang).

https://doi.org/10.1016/j.apor.2022.103106
Received 18 September 2021; Received in revised form 8 February 2022; Accepted 11 February 2022
Available online 17 March 2022

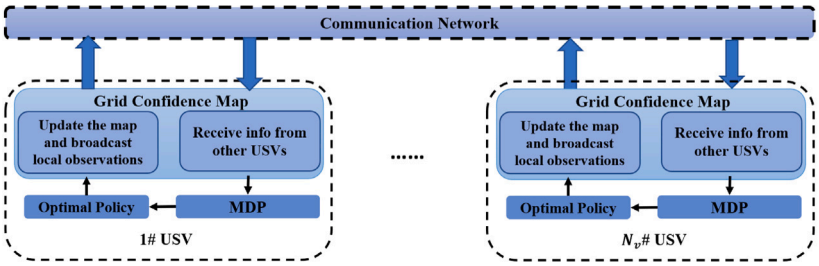


Fig. 1. The framework of the proposed USV coordination algorithm.

could be marine animals or survivors floating on the ocean surface. In this application context, the conventional formation control strategies are not very efficient for detecting object targets and acquiring environmental information. Due to the requirement of maintaining a formation, individual USVs share similar search trajectories, which results in resembling angles and positions for observing an object target. An improved design is to remove the constraint of maintaining the formation and coordinate USVs to observe object targets from various perspectives. Therefore, it is expected that the USV fleet can obtain more comprehensive information about the object targets and improve the search efficiency compared to formation control strategies.

Reinforcement learning (RL) algorithms are widely implemented in the field of artificial intelligence (AI). For instance, AlphaGo (Silver et al., 2016), an AI robot based on RL methods, defeated a couple of best professional human players in the game of Go. An RL algorithm models interactions between an agent and the environment, usually framed as a Markov decision process (MDP): an agent receives rewards by performing actions, and the goal of the agent is to take actions that maximize the cumulative reward (Sutton and Barto, 2018). An MDP framework is suitable for modeling behaviors of a single USV: an agent could be considered as a USV aiming to find object targets. By appropriately defining reward functions, the USV is driven to choose actions that are beneficial to achieve the predefined objective.

In this article, a USV coordination algorithm based on RL methods is presented. Central to the proposed coordination algorithm is the design of a grid confidence map. As demonstrated in Fig. 1, the local sensing information of individual USVs is shared with every fleet member through the wireless communication network. Based on both received and self-perceived information, a USV constructs a grid confidence map, which reflects how much information the fleet knows about every region in the search area. It is worth mentioning that each USV maintains its grid confidence map. A better design is to unify the diverse grid confidence maps and generate a uniform map that applies to all fleet members. However, considering the limited communication bandwidth in the ocean environment, transiting local observations is more applicable than unifying multiple grid confidence maps. In the future, we will improve this design by extending the communication bandwidth and adding the information fusion algorithm to produce a fused grid confidence map.

Additionally, the distributed USV coordination architecture is employed, where each USV in the fleet has the autonomy to plan its search trajectories. Search behaviors of a single USV are governed by an MDP, where reward functions are defined based on the constructed grid confidence map. As a result, USVs are encouraged to explore new areas to search object targets and prevented visiting already searched locations repeatedly. The optimal policy, i.e., search routes, of individual USVs is solved via a policy-iteration based path planning algorithm. After a USV performs an action, it will update the grid confidence map with the new observation and broadcast the local observations to other USVs. Besides, inter-vehicle collisions are avoided by specifying policy constraints. Results from real-world experiments demonstrate that compared to the conventional formation control strategy and the uncoordinated algorithm, our method is more efficient and intelligent for searching object targets in the ocean environments.

2. Related works

The majority of recent researches on the multi-USV coordination problem mainly focuses on formation control strategies, in which USVs are commanded to maintain a desired formation during the operations. Commonly used formation control strategies include the leader–follower methods (Shojaei, 2015; Sun et al., 2020), virtual structure methods (Do, 2011), behavior-based methods (Glotzbach et al., 2015), and graph-based methods (Liu et al., 2018). Despite the outward differences, these approaches are essentially similar: they treat the formation as a unified structure, while members in the formation are forced to maintain designated relative positions.

In our application, the objective of the USV fleet is to search and observe object targets. Formation control strategies are not ideal in this scenario since members in the formation share resembling observation angles and positions, which impedes the acquisition of the comprehensive information about object targets. An improved design is removing the constraint of maintaining a formation, where each individual USV has the autonomy to self-plan its trajectory depending on the current search situation. This idea can be framed as the swarm control strategies (Tan and Zheng, 2013), where members in a multi-agent system are controlled separately rather than being treated as an entire formation.

Swarm control strategies have been widely implemented in the field of multi-USV coordination problems. Qin et al. (2017) presented a multi-USV swarm control algorithm based on the hierarchical control strategy. In their work, the USV fleet is controlled to proceed to the predefined target positions while avoiding collisions. To coordinate USV behaviors, they employed a flocking strategy to control the distance variance between each USV and the fleet's center point. Particle swarm optimization (PSO) is another commonly used approach in coordinating multi-USV systems (Guo et al., 2020; Xin et al., 2019). For instance, Xia et al. (2020) proposed a local path planning algorithm based on PSO, which produces optimal USV trajectories to avoid dynamic obstacles during the USV sailing. Many other swarm control methods, such as artificial potential field (APF) (Tan et al., 2020a) and fast marching square (FMS) (Tan et al., 2020b) algorithms, can also be implemented in the USV coordination problem.

In terms of RL-based applications for USVs, most research concentrates on designing a controller to guide a USV following a predefined path. For instance, Woo et al. (2019) proposed a deep reinforcement learning (DRL) based controller for USV path tracking, where a USV achieves a self-learning capability using the deep deterministic policy gradient (DDPG) algorithm to follow a guidance trajectory. Jin et al. (2019) presented a USV motion control method based on the actor–critic scheme, where a USV is controlled to follow a trajectory in complex maritime environments. Zhao et al. (2020) proposed a path following controller to navigate a USV following a trajectory based on DRL methods. In their work, to reduce the complexity of the control law, an improved deep Q network (DQN) is designed to make control decisions based on USV states.

For applying RL algorithms on multi-USV systems, limited research work has been carried out. Zhou et al. (2019) presented a formation

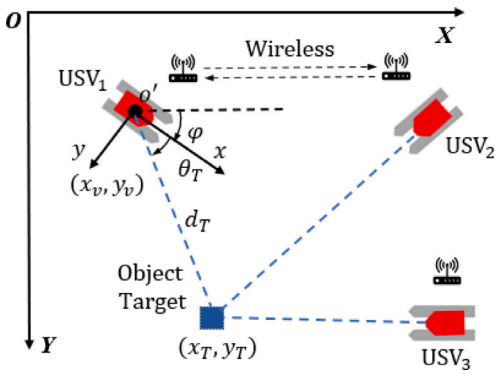


Fig. 2. Demonstrations of the USV fleet perceiving the object target in the proposed collaborative search algorithm. In the diagram, XOY is the global frame coordinate, and $xo'y$ is the body frame coordinate. Positions of a USV and an object target are (x_v, y_v) and (x_T, y_T), respectively, and the USV heading is φ. The distance and the angle difference between the USV and the target are d_T and θ_T, respectively.

control strategy for coordinating USVs based on DRL methods. They employed the DQN framework to produce USV actions (i.e., discretized heading commands), in which multiple sub-reward functions are designed to stipulate USV behaviors such as avoiding collisions and maintaining the formation. Liu et al. (2020) also adopted the DQN to generate USV actions and coordinate a USV fleet searching for underwater targets. In their method, multiple probability maps are constructed to represent environmental information, such as target locations, USV communication ranges, and obstacle positions. Then, map information is utilized to define reward functions, where USVs are encouraged to find underwater targets and avoid colliding with obstacles.

By summarizing these works, it can be discovered that applying RL methods to the multi-USV coordination problem is still in its infancy and requires further research. Besides, the majority of the aforementioned works were evaluated in simulated environments. Real-world implementations of RL-based USV coordination algorithms are rare in recently published research articles. Motivated by these considerations, we propose a multi-USV coordination algorithm based on RL methods and implement it on USVs in the real-world ocean environment to evaluate its validity.

The remaining of the article is organized as follows. Section 3 presents the overview of the proposed coordination algorithm; Section 4 presents the modeling procedure, including constructing the grid confidence map; Section 5 demonstrates the planning procedure, where multiple sub-reward functions are defined and a policy-iteration based path planning algorithm is presented; Section 6 shows the experiment setup and results.

3. An overview of the proposed method

In this work, the goal of the USV fleet is to search multiple object targets over the surface of an unknown water region without obstacles. To mimic floating objects on the ocean surface, object targets contain both mobile and stationary floating objects, and their positions are clouded to USVs.

Fig. 2 demonstrates how the USV fleet detects an object target with the proposed USV coordination algorithm. Assume that there are N_v USVs in the fleet and N_T object targets in the search area. To perceive the environment and communicate with other USVs, a single USV is equipped with three main sensory modules: positioning, communication, and object detection modules. At every time step, positioning sensors measure USV's positions $p_v = (x_v, y_v)$ and headings φ in the global frame; the wireless communication module realizes information exchanges with other USVs; the object detection sensor detects object targets if they are within the perception range of a USV. In this work, an onboard camera is employed as the object detection sensor, which captures images of object targets. Then, the onboard computer processes the captured images to generate the distance and angle difference between the USV and the object target, denoted as d_T and θ_T, respectively. Thus, the object target position $p_T = (x_T, y_T)$ in the global frame can be calculated based on the USV's position and heading information:

$$\begin{cases} x_T = x_v + d_T \cos\left(\varphi + \theta_T\right) \\ y_T = y_v + d_T \sin\left(\varphi + \theta_T\right). \end{cases} \tag{1}$$

Details of onboard sensors can be found in Section 6.1.

The proposed coordination algorithm comprises two principal procedures: modeling and planning. In the procedure of modeling, we define the grid confidence map to indicate how much information the fleet knows about every region in the search area. This information refers to the confidence of the fleet in believing whether a region contains object targets. Each USV in the fleet will maintain a grid confidence map based on local observations and other USVs' perceptions received via a wireless communication network. Search behaviors of a single USV are modeled by an MDP, where elements in the MDP framework are adapted to fit the coordination problem: states are defined as USV positions in the search area; actions are considered as possible moving directions of a USV; policies can be treated as USV search routes.

In the planning procedure, optimal search routes are determined. In an RL algorithm, reward functions define agent behaviors and stipulate how we want the agent to accomplish its objective. In this work, USVs are expected to search object targets cooperatively and achieve optimal search efficiency. To achieve this goal, various types of sub-reward functions are defined based on the information from the grid confidence map so that USVs are encouraged to explore new regions (i.e., areas with low confidence) and prevented visiting already searched areas repeatedly. The policy-iteration algorithm is adapted to solve for optimal policies, i.e., search routes, where policy constraints are applied to avoid inter-vehicle collisions among the USV fleet.

4. Modeling

4.1. Search area

For the computational feasibility, the search area is modeled as a grid with M cells in a row and N cells in a column as presented in Fig. 3. The size of a cell is defined as $L_x \times L_y$, where L_x and L_y are the length and width of a cell, respectively. A cell can be referenced by its index, i.e., C_i where $i \in [1, MN]$. A vector $\mathbf{C} = [C_1, C_2, \ldots, C_{MN}]$ is defined to store cell indexes. Besides, C_i can also represent the position of a cell, such that $C_i = (x_i, y_i)$ is the center point of a cell C_i. Object targets, including both mobile and stationary, are placed inside the search area. Positions of object targets are clouded to USVs.

4.2. Grid confidence map

The grid confidence map is a mathematical model that reflects the current knowledge of the USV fleet on the search area. In a nutshell, key concepts of the grid confidence map are listed below:

- The map is constructed based on the gridded search area (i.e., Fig. 3), where each cell has a value (between 0 and 1), termed *cell confidence*, indicating how confident the fleet believes a cell containing object targets;
- The map is generated by a USV based on both self-perceived information and observations from other USVs received via a communication network;
- The map information is used to direct future movements of individual USVs in the planning procedure.

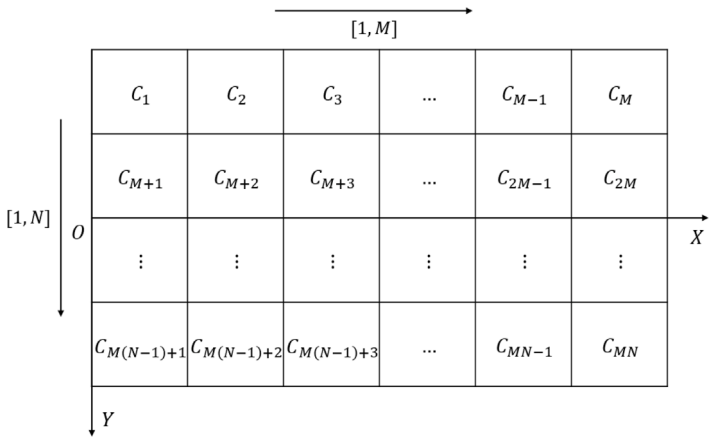


Fig. 3. Model the search area as a grid to represent positions and cells.

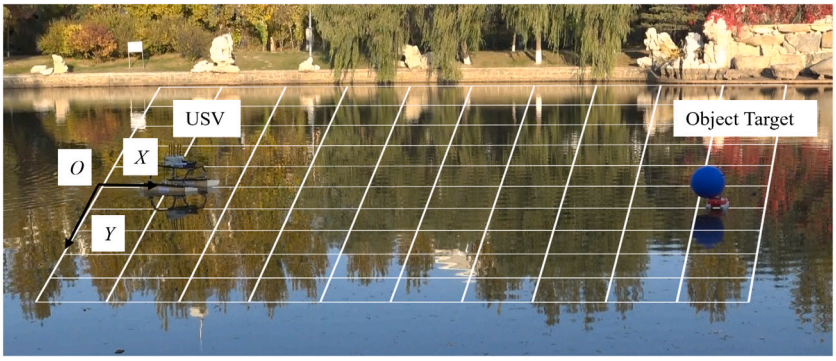


Fig. 4. The setup of the water test for determining the expression of cell confidence. The USV position is located at $(1, 0)$ m, and the USV heading is toward the positive direction of the X axis. A blue balloon is employed as the object target, and its position is varying in the search area.

4.2.1. Definition of cell confidence

Let us define b_i^j as the cell confidence of a cell C_i calculated by the USV_j, where $i \in [1, MN]$ and $j \in [1, N_v]$. Recall that the cell confidence indicates how much information the fleet knows about C_i. This information refers to whether C_i contains object targets, which is perceived by the object detection sensor. If the object detection sensor on USV_j can accurately perceive the cell C_i, the value of b_i^j is expected to be high; on the other hand, the b_i^j value will be low if the object detection sensor cannot properly observe the cell C_i. Therefore, the performance of the object detection sensor is related to the value of b_i^j.

A water test is conducted to exam the performance of the implemented object detection sensor. Specifically, as presented in Fig. 4, a USV is placed stationary at $(1, 0)$ m in a 10×10 m^2 search area with 10×10 cells. An object target, which is a blue balloon (radius is 40 cm), is randomly placed at different positions in a cell with N_{total} times ($N_{total} = 100$ in the test). The USV observes the object target's position via the onboard object detection sensor (i.e., a camera). After the USV captures the object target images, a support vector machine (SVM) is employed to transfer the pixel information of the object target to the actual position in the search area. If the observed position is within the same cell as the actual object target, this observation is considered as valid. The number of valid observations, i.e., N_{valid}, is recorded, and the cell confidence b_i^j is defined as the success rate of valid observations in a cell:

$$b_i^j = \frac{N_{valid}}{N_{total}}. \tag{2}$$

From (2), it can be deduced that the higher the value of cell confidence, the USV is more confident about the information perceived from a cell.

The model of onboard camera is Raspberry PI Camera Module V2 (Element14.com, China). The object detection algorithm is based on YOLOv2, where the detailed algorithm can be found in Redmon and Farhadi (2017). In our tests, we found that the implemented object detection algorithm was accurate and robust. The accuracy rate is 92% in detecting the floating blue balloon. This accuracy rate is calculated from experiment results. In this experiment, the object detection algorithm aimed to identify a blue floating balloon from images collected from the real-world environment (see Fig. 5). The identification result was compared with human labels, and the implemented object detection algorithm correctly found the blue balloon in 214 photos out of 232 photos, i.e., achieving 92% accuracy rate. It should be mentioned that LiDAR sensor can also be used as an object detection sensor, but considering the high cost of LiDAR sensors, we used a camera to detect floating objects in this work.

Fig. 5 demonstrates images captured from the onboard object detection sensor (i.e., camera) during the water test, where the values of distance and angle difference between the camera and object target, i.e., d_T and θ_T, obtained after the image processing procedures are labeled beneath the detected object target. It should be noted that these images are captured from a fixed position object target. However, due to the environmental disturbances, such as winds and waves, the observed object target positions are different. Possible approaches to mitigate errors produced by the environmental disturbances include adding more object detection sensors and attaching a sophisticated tripod head on the camera to overcome the disturbances.

By calculating b_i using (2) over all cells in the search area, a grid confidence map is obtained. Table 1 presents the calculated grid confidence map, and Fig. 6(a) shows the distribution of this map over the search area. It can be observed that cell confidence values are high for cells near the center of the USV perception area. This phenomenon

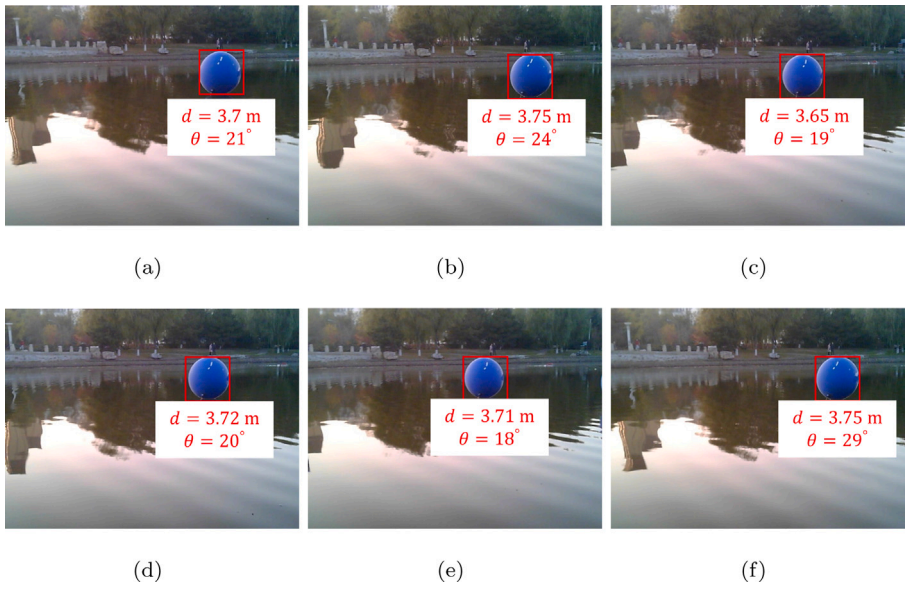


Fig. 5. Captured images of the object target by the onboard object detection sensor, i.e., an onboard camera. After the image processing, the distance and angle difference between the camera and the USV, i.e., (d_T, θ_T), are (a) $(3.7\text{ m}, 21°)$; (b) $(3.75\text{ m}, 24°)$; (c) $(3.65\text{ m}, 19°)$; (d) $(3.72\text{ m}, 20°)$; (e) $(3.7\text{ m}, 18°)$; (f) $(3.75\text{ m}, 29°)$.

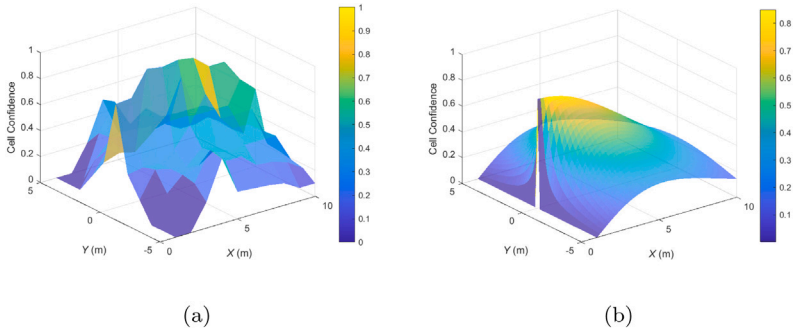


Fig. 6. (a) The calculated grid confidence map based on results from the water test. (b) The plot of the selected general function. For these two diagrams, the horizontal plane is the search area, and the vertical axis indicates values of cell confidence for each cell in the search area.

Table 1
The grid confidence map calculated from the water test results; The unit for the X and Y axis is meter.

Y \ X	1	2	3	4	5	6	7	8	9	10
−5	0	0	0.2	0.5	0.21	0.2	0.19	0.17	0.1	0.1
−4	0	0	0.3	0.52	0.4	0.4	0.3	0.29	0.25	0.17
−3	0	0.21	0.32	0.35	0.43	0.39	0.41	0.41	0.41	0.19
−2	0.15	0.35	0.32	0.3	0.41	0.38	0.49	0.49	0.4	0.21
−1	0.3	0.32	0.31	0.39	0.54	0.75	0.52	0.52	0.52	0.22
0	0.8	0.81	1	0.98	1	0.98	0.88	0.88	0.71	0.6
1	0.3	0.34	0.31	0.41	0.52	0.79	0.46	0.46	0.46	0.21
2	0.15	0.32	0.27	0.31	0.37	0.4	0.4	0.4	0.39	0.2
3	0	0.2	0.33	0.32	0.43	0.37	0.37	0.37	0.37	0.21
4	0	0	0.32	0.53	0.32	0.41	0.31	0.3	0.3	0.2
5	0	0	0.2	0.5	0.21	0.2	0.19	0.17	0.1	0.1

is due to the characteristic of the implemented object detection sensor, i.e., the onboard camera: pictures of object targets will be more explicit when they are captured near the camera's central view. Thus, the calculated object target positions will be more reliable in the center areas than the side regions of the USV perception area.

To model this feature of the onboard object detection sensor, a general function that fits the distribution of the calculated grid confidence map is defined. Fig. 6(b) shows the plot of the selected general function, and its expression can be presented as:

$$b_i^j(t) = \alpha \cos(\theta_i^j) e^{-(d_i^j/r_s)^2} + n_i(t), \tag{3}$$

where α is the coefficient derived from the distribution of the calculated grid confidence map; d_i^j and θ_i^j are distance and angle difference between the USV$_j$ and cell C_i, respectively; r_s is the perception range of the object detection sensor; $n_i(t)$ represents the environmental noises, which are modeled by the white Gaussian noises with zero mean and σ^2 variance. Based on water test results, we set $r_s = 5$ m; $\alpha = 0.85$; $\sigma = 0.02$. The difference between the real and generated grid confidence maps is 0.026 per cell.

It should be mentioned that the selection of the general function is not unique. The purpose of defining the general function is to mathematically calculate the cell confidence based on the implemented object detection sensor's characteristics. As long as a function produces a

similar distribution as the obtained grid confidence map (i.e., Fig. 6(a)), it can also serve this purpose and be selected as the general function.

4.2.2. Construct the grid confidence map

During the fleet maneuver, a USV constructs a grid confidence map based on both local observations and the sensing information received from other USVs.

Suppose that there are N_v USVs in the search area. At time t, a USV calculates cell confidence values for cells covered by its perception area via (3). Besides, through the communication network, this USV receives cell confidence values calculated by other USVs based on their perception areas. Both self-perceived and received information are fused to construct the grid confidence map. For areas that are not perceived of any USVs, cell confidence values will decrease since the utility of the perceived information attenuates over time. A cell is considered as covered by a USV's perception area if the distance between its center point and the USV position is within the perception range of the object detection sensor, i.e., $|C_i - p_v| < r_s$.

Denote the vector $\mathbf{B}(t) = [b_1(t), b_2(t), \ldots, b_{MN}(t)]$ as the grid confidence map on a USV at time t. An element in $\mathbf{B}(t)$, i.e., $b_i(t)$ where $i \in [1, MN]$, can be represented as:

$$b_i(t) = \begin{cases} 1 - \prod_{k=1}^{N_v} \left(1 - b_i^k(t)\right) & \text{if } C_i \in W_{OA} \\ \tau b_i(t-1) & \text{if } C_i \notin W_{OA} \end{cases}, \tag{4}$$

where $b_i^k(t)$ is the cell confidence of a cell C_i in the perception area of USV_k at time t calculated via (3); W_{OA} represents the set of cells covered by the overall perception area of all USVs; τ is the attenuation rate applied on cells that are not detected by any USVs.

Notice that, due to the existence of mobile object targets, the timeliness of cell confidence is critical: old cell confidence cannot timely reflect the information of mobile object targets. Thus, we attenuate history cell confidence values for cells not detected by USVs to reduce the significance of the previously perceived information. Due to the similar reasons, we do not use history values, i.e., $b_i(t-1)$, to update the cell confidence for cells perceived by USVs. In general, it is preferable to use new observations overwriting the history cell confidence values in our application since the new sensor observations are more instructive than the history information. In implementations, confidence values of all cells are initialized as 0.1 with the attenuation rate $\tau = 0.99$. We have tried different combinations of initial confidence values and attenuation rates. We found that different initial confidence values will not have a significant effect on search performance. Since we define the confidence threshold is 0.5 (this value is determined since the confidence value varies from 0 to 1; thus, we choose a middle value as the threshold), the initial confidence value should be a small value, e.g., 0.1. On the other hand, we found that a low attenuation value (e.g., $\tau = 0.1$) results a diverge of the confidence map. In this case, the confidence value of a cell drops quickly, and the USV fleet cannot maintain a high confidence value over the entire search area. Considering the above reason, the implemented attenuation rate is a large value, i.e., $\tau = 0.99$.

4.3. Model behaviors of a single USV as an MDP

4.3.1. RL basis

An RL problem is usually framed as an MDP, which comprises a tuple $(S, \mathcal{A}, P, r, \gamma)$ (Sutton and Barto, 2018):

- S is a state space;
- $\mathcal{A}$ is an action space;
- P are state transition probabilities between states;
- r is the reward function defined on the transitions;
- γ is the discount factor.

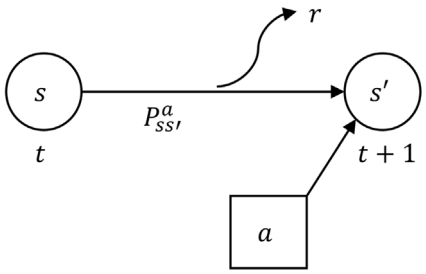


Fig. 7. A basic MDP model.

Fig. 7 shows a basic MDP model. At time t, the agent is in the state (s, $s \in S$), and after performing an action (a, $a \in \mathcal{A}$), it transmits to a new state (s', $s' \in S$) according to the state transition probability $P^a_{ss'} = \mathbb{P}[s_{t+1} = s' | s_t = s, a_t = a]$ and receives a reward r. A policy π is used to select actions, and the agent uses its policy to interact with the environment to obtain a trajectory of states, actions, and rewards, i.e., $h_{1:T} = s_1, a_1, r_1 ..., s_T, a_T, r_T$. The return G_t is the total discounted reward from time-step t onward, i.e., $G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2}, \ldots = \sum_{k=t}^{\infty} \gamma^{k-t} r_k$, where $\gamma \in [0, 1]$ is the discount factor. The agent's goal is to find an optimal policy that maximizes the cumulative discounted reward from the start state. The remaining section presents the method that adapts elements of an MDP to model a single USV search behaviors.

4.3.2. State space

For the MDP of a single USV path planning problem, we define the state space containing possible USV's positions in the search area. To reduce the computation, the USV's position is simplified as the position of the cell where it occupies. This approximation is acceptable since the size of a USV is comparable to the size of a cell and the resolution of positioning measurements is less than the size of a cell. In general, a state can be represented as $s = C_i$ where $i \in [1, MN]$, and the state space $S = [C_1, C_2, \ldots, C_{MN}]$ contains all cells in the search area.

4.3.3. Action space

The action space comprises possible actions that an agent could select. In this work, we choose the movement directions as USV's actions. It should be mentioned that the surge speed of a USV is excluded in the action space since the USV speed is adapted based on the distance to the USV's destination: a USV will accelerate when it is far from the destination or decelerate when the destination is approaching.

As presented in Fig. 8(a), a USV can select one of five actions (i.e., five arrows), ranging from $-60°$ to $60°$ with increments of $30°$. Such a design is compliant with the kinematic characteristics of USVs, i.e., a USV can only make a turn within a certain range during one control step (Liu et al., 2017). Notice that selecting the angle range $[-60°, 60°]$ is based on our previous water test results on a single USV (Miao et al., 2019), which is also employed as the robotic platform in this work. This range can be modified for implementing the proposed algorithm on other robots. An action is denoted as the movement direction. For instance, an action $a = -30$ represents that the USV moves toward $-30°$ direction. Therefore, the action space can be represented as $\mathcal{A} = \{a_1 = -60, a_2 = -30, \ldots, a_5 = 60\}$.

The USV perception area is evenly separated into six sectors with five actions. Denote w_1, w_2, ..., w_6 as sets of cells covered by six sectors. A cell is considered in a sector if its center point is inside a sector. The size of an action step is equivalent to the perception range of a USV, i.e., r_s. During the course of performing an action, two adjacent sectors alongside the USV movement are defined as the closely detection area, denoted as W_a. For instance, if a USV chooses the action $a_1 = -60$, the combination of w_1 and w_2 is defined as the closely detection area (i.e., $W_{-60} = w_1 + w_2$). This design is motivated

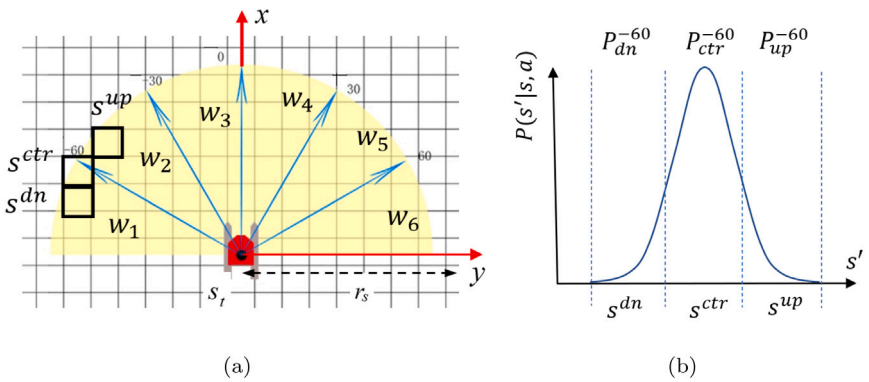


Fig. 8. The action space and state transition probabilities. (a) The action space, where arrows indicate possible actions that a USV can take. Possible states that a USV could arrive after performing the action $a = -60$ are highlighted with black squares. (b) The normal distribution of state transition probabilities, where the horizontal axis represents the new state s_{t+1} and the vertical axis represents the state transition probability.

by the characteristics of the onboard object detection sensor, which captures more explicit pictures of object targets when they are near the perception area center. Thus, the information perceived from the closely detection area is more reliable and informative compared to other sectors. For each action, the closely detection area is defined as: $W_{-60} = w_1 + w_2$, $W_{-30} = w_2 + w_3$, …, $W_{60} = w_5 + w_6$.

4.3.4. State transition probability

Due to the existence of environmental disturbances, such as winds and waves, the state transition process is stochastic in this work.

Previous water test data (Miao et al., 2019) reveals that a USV could reach one of three locations (i.e., states) after performing an action. There is a high probability that the USV would arrive at the desired cell, but it is also possible that the USV could drift to either up or down cells. This state transition process follows a Gaussian model. For instance, Fig. 8(a) highlights three possible states (i.e., s^{dn}, s^{ctr}, and s^{up}) after a USV taking the action $a_1 = -60$, and Fig. 8(b) presents the state transition probabilities of the corresponding state transition process. It can be observed that a USV has a high probability (i.e., P_{ctr}^{-60}) to enter the desired state (i.e., $s' = s^{str}$), but it is also possible that the USV could drift to either up and down cells (i.e., $s' = s^{up}$ and $s' = s^{dn}$) with state transition probabilities P_{up}^{-60} and P_{dn}^{-60}, respectively. This state transition process also applies to other actions. In general, the state transition probability of an action can be represented as:

$$P_{ss'}^{a} = \begin{cases} P_{dn}^{a} & \text{if } s' = s^{dn} \\ P_{ctr}^{a} & \text{if } s' = s^{ctr} \\ P_{up}^{a} & \text{if } s' = s^{up} \end{cases}. \tag{5}$$

For an action, P_{dn}^{a}, P_{ctr}^{a}, and P_{up}^{a} are defined as 0.1, 0.8, and 0.1, respectively in implementations. Thus, the USV has a high probability of entering the correct cell after performing an action and has marginal probability of entering a wrong cell.

5. Planning

5.1. Define reward functions

Reward functions in an RL problem should be designed to enforce an agent to learn the desired behavior and complete the task as expected. For navigating a USV in the fleet and realizing the optimal coordination, a set of sub-reward functions are designed to stipulate USV behaviors.

5.1.1. Object target reward (r_T)

The object target reward is designed to encourage a USV to search object targets. Assume that a USV detects n_T object targets after performing an action a; then, the object target reward r_T can be represented as:

$$r_T = n_T. \tag{6}$$

It should be noted that the object target reward also applies for mobile object targets. Once a mobile object target changes its position, the detection of this object target will be treated as the new detection and increase the object target reward. As a result, the USV will choose the actions to chase the mobile object target to obtain the reward.

5.1.2. Information reward (r_I)

The information reward stimulates a USV to explore new regions, i.e., areas with low confidence values on the grid confidence map. To mathematically measure how much information a USV can obtain by visiting an area, the theory of information entropy (Shannon, 2001) is employed to define the information reward.

Denote $\Delta I_i(t)$ is the change of information entropy after a USV visiting a cell C_i, which can be calculated by:

$$\Delta I_i(t) = |\log b_i(t) - \log b_i(t-1)|, \tag{7}$$

where $b_i(t)$ and $b_i(t-1)$ are confidence values of cell C_i at time t and $t-1$, respectively. A high value of ΔI_i indicates that the fleet's knowledge of cell C_i will be significantly improved if a USV searches the cell C_i.

The information reward is defined as the sum of ΔI for cells covered by the closely detection area of an action. For instance, the information reward of the action $a_1 = -60$ is calculated as the sum of ΔI for cells in W_{-60}, i.e., $r_I = \sum_{C_i \in W_{-60}} \Delta I_i(t)$. In general, the information reward r_I can be presented as:

$$r_I = \sum_{C_i \in W_a} \Delta I_i(t). \tag{8}$$

5.1.3. Repeat search penalty (r_P)

The repeat search penalty is designed to prevent the USV repeatedly searching the same area. This penalty is defined as the frequency of a USV detecting the same area, which can be expressed as:

$$r_P = -\frac{N_P}{\Delta T_P}, \tag{9}$$

where N_P is the number of cells in the USV perception area that have been detected before and ΔT_P is the time interval between two repetitive searches. Note that, if a cell is in the USV perception area for the first time, the count of N_P will not increase until the cell is out of the perception area. The time of the cell leaving the perception area will be recorded for computing the value of ΔT_P.

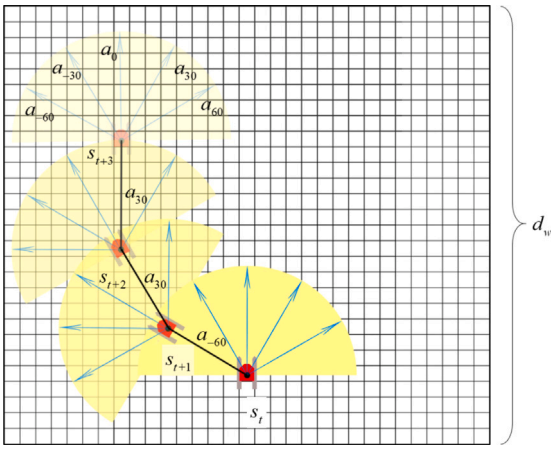


Fig. 9. A sliding window. The USV plans search path inside the sliding window with finite time steps to reduce the computational load. In the presented sliding window, the USV plans a search trajectory within the next three time steps.

5.1.4. Boundary penalty (r_B)

The boundary penalty aims to avert a USV leaving the search area, which can be defined as:

$$r_B = -N_B, \tag{10}$$

where N_B is the number of boundary cells that a USV detects. A boundary cell is defined as a cell locating at the boundaries of the search area. With the boundary penalty, a USV will avoid choosing actions that direct it to approach the search area's boundaries. Notice that, the search area can also be irregular thanks to the boundary penalty.

In general, the total reward is defined as the sum of all aforementioned sub-rewards:

$$r = \lambda_T \cdot r_T + \lambda_I \cdot r_I + \lambda_P \cdot r_P + \lambda_B \cdot r_B \tag{11}$$

where λ represents the coefficient of different types of sub-reward functions. We can adjust values of λ to specify the significance of the corresponding USV behaviors. In implementations, we set $\lambda_T = 0.2$, $\lambda_I = 0.3$, $\lambda_P = 0.2$, and $\lambda_B = 0.3$.

5.2. Generate the optimal policies

5.2.1. Sliding window

Considering the finite perception range of a USV and the limited onboard computational resources, a sliding window is defined to reduce the computation of search routes. As presented in Fig. 9, the sliding window is a square region covering the front area of a USV. At every time step, instead of computing the search path over the entire search area, the USV only plans the future movements inside the sliding window. Thus, in the planning procedure, the state space shrinks to contain all cells in the sliding window, i.e., W_o, where W_o is the set of cells covered by the sliding window. This design allows the USV to quickly calculate a search route and timely respond to new observations. The length of the sliding window d_w relates to the USV kinematics, which is calculated as:

$$d_w = N_w v \Delta T, \tag{12}$$

where N_w is an adjustable coefficient that controls the scale of the slide window; v is the surge speed of a USV; ΔT is the time interval between two consecutive decision making processes.

5.2.2. Path planning

After reward functions are determined, search routes of individual USVs are generated. The policy-iteration algorithm is employed to solve for the optimal policy, i.e., search routes, which comprises two main steps, namely policy evaluation and policy improvement.

In the step of policy evaluation, action-value functions (i.e., Q-functions) are estimated. To achieve that, we first calculate state-value functions for a given policy π based on the Bellman expectation equation (Sutton and Barto, 2018):

$$V_\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s)(r + \gamma \sum_{s'\in S} P^a_{ss'} V_\pi(s')), \tag{13}$$

where $V_\pi(s)$ is the state-value function. According to the defined state transition probability (5) and reward functions (11), we can rewrite (13) as:

$$\begin{aligned} V_\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s) \{ r + \gamma [& P^a_{dn} V_\pi(s^{dn}) + \\ & P^a_{ctr} V_\pi(s^{ctr}) + P^a_{up} V_\pi(s^{up})] \}. \end{aligned} \tag{14}$$

In the above equation, the discount factor (i.e., γ) is defined as 0.9. The larger the γ is, the broader region a USV considers in planning the search route. Under the current policy π, the state-value function is iteratively calculated via (14) until converge, i.e., the maximal update of the state-value function (i.e., Δ) is less than the convergence tolerance (i.e., ς). To balance the tradeoff between the algorithm performance and the processing time, the converge tolerance is selected as 10^{-6} in experiments to obtain a well-algorithm performance and save the processing time. Then, for every state and action pair, the action-value function is calculated based on the converged state-value function via:

$$Q_\pi(s,a) = r + \gamma [P^a_{dn} V_\pi(s^{dn}) + P^a_{ctr} V_\pi(s^{ctr}) + P^a_{up} V_\pi(s^{up})]. \tag{15}$$

In the step of policy improvement, the policy is updated with respect to the obtained action-value function. The most common approach to update the policy is the greedy maximization of the action-value function, i.e., $\pi(s) = \arg\max_{a\in A} Q_\pi(s,a)$. However, to ensure the exploration, we adopt the ϵ-greedy algorithm to update the policy:

$$\pi(a|s) = \begin{cases} 1-\epsilon & \text{if } a = \arg\max\limits_{a\in A} Q_\pi(s,a) \\ \epsilon & \text{if } a \neq \arg\max\limits_{a\in A} Q_\pi(s,a) \end{cases}, \tag{16}$$

where ϵ is the probability of choosing a random action a that does not make the Q-function maximal. The value of ϵ is selected as 0.2 in implementations.

Algorithm 1 presents the pseudo-code for the policy-iteration based path planning algorithm. First, state-value functions of all states in the sliding window $s \in W_o$ are initialized as 0, and the action policy is randomly initialized. Then, in the step of policy evaluation, a convergence tolerance ς is set, and the state-value function keeps updating via (14) until converge, i.e., $\Delta < \varsigma$. Next, the old policy is updated via the ϵ-greedy algorithm in the step of policy improvement. Once the policy is converged, the planning process terminates and outputs the optimal policy π^*; otherwise, the algorithm returns to the policy evaluation step and repeat the steps.

It should be mentioned that the optimal policy generated from Algorithm 1 is based on the current grid confidence map and is not permanent. In every time step, the grid confidence map is updated with new USV observations via (4). As a result, reward functions, which are defined based on the information from the grid confidence map, vary in every time step. A new optimal policy will be calculated via Algorithm 1 based on new reward functions. The new policy overwrites the old one, allowing a USV to timely adjust search behaviors to fit the new observed information. In general, the overall search trajectory of an individual USV is a sequence of optimal policies generated from varying reward functions. Additionally, the number of states (i.e., cells) in the

Algorithm 1 Policy-Iteration Based Planning Algorithm

Initialize value function $V_\pi(s)$ and policy $\pi(s)$ for all states in the sliding window, i.e., $s \in W_o$
Policy Evaluation
Set the convergence tolerance ς
while $\Delta \geq \varsigma$ **do**
  $\Delta = 0$
  for $s \in W_o$ **do**
    $v = V_\pi(s)$
    Calculate $V_\pi(s)$ via (14)
    $\Delta = \mathbf{max}(\Delta, |v - V_\pi(s)|)$
  end for
end while
Policy Improvement
policy-stable = True
for $s \in W_o$ **do**
  old-action= $\pi(s)$
  Update the policy $\pi(s)$ via ϵ-greedy algorithm
  if *old-action* $\neq \pi(s)$ **then**
    policy-stable = False
  end if
end for
if *policy-stable* **then**
  Stop and return the optimal policy π^*
else
  Go to Policy Evaluation, i.e., Step 2.
end if

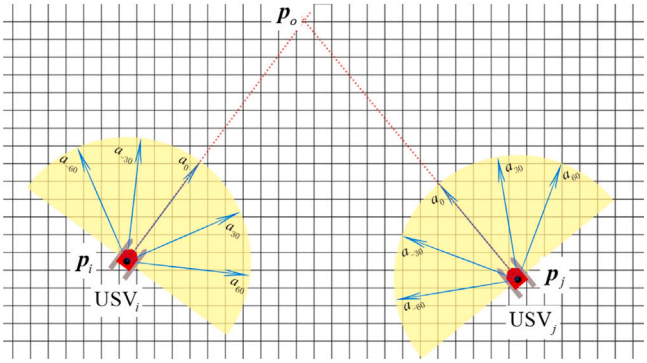


Fig. 10. The search conflict between two USVs, where the yellow half circle represents the perception area of a USV. p_i and p_j are USV positions, and p_o is the intersection of the prospective trajectories of two USVs.

planning procedures is significantly reduced due to the implementation of sliding windows. Thus, the training time is neglected in the planning procedure.

5.2.3. Policy constraints

Policy constraints are designed to avoid inter-vehicle collisions. When two USVs are very close to each other and have the potential to collide, i.e., future trajectories overlap at an intersection point, the USV with the further distance to the intersection point will wait until the other USV completes its actions. For example, in Fig. 10, search trajectories of USV$_i$ and USV$_j$ overlap at point p_o, and the distance between them is less than the threshold, i.e., $|p_i - p_j| < D_{th}$. In this case, the policy constraints will interfere to prevent the collision: USV$_j$ will stop and wait since it is further to the intersection point p_o compared to USV$_i$.

6. Experiment

6.1. USV hardware

In the experiment, five identical USVs were developed as the robotic platform to implement the proposed USV coordination algorithm. Fig. 11 presents the hardware architecture and the configuration of components on a single USV. Major components include an onboard computer (i.e., Raspberry PI 3), an autopilot (i.e., Pixhawk 2.4.6 autopilot), two communication modules (i.e., Xbee Pro538), a global positioning system (GPS) module, and an onboard camera with the image processing board (i.e., Odroid-XU4 development board). The size (length×width×height) of a USV is $56 \times 27 \times 26$ cm^3, and the weight is 2.5 kg.

On a single USV, the proposed coordination algorithm runs on the onboard computer, which generates search routes (i.e., a series of waypoints) based on local observations and the information received from other USVs. The autopilot is responsible for realizing the planned search routes, which produces pulse width modulation (PWM) signals to control the speed of two thrusts. As a result, the USV is controlled to proceed toward waypoints sequentially. Two communication modules are employed, namely Xbee-L and Xbee-G modules. Two modules operate at different communication frequencies, where the Xbee-L module is used to communicate with other USVs in the fleet and the Xbee-G module is employed to transmit USV statuses to the ground station for the monitoring purpose. The maximal transmission distance of a Xbee module is 800 m. In case of emergencies, such as mechanical failures and recovering the USVs after the test, a USV can be manually controlled via a transmitter on the shore.

6.2. Ocean tests and results

To evaluate the performance of the proposed coordination algorithm, multiple ocean tests were conducted at the Zhuhai city shore, China in August 2020. Test results are analyzed and compared with the traditional coordination algorithm, i.e., the formation control strategy, and the uncoordinated control algorithm.

6.2.1. Ocean test setup

Fig. 12 demonstrates the experiment field of the ocean tests. The search area is a square water region with the size of 100×100 m^2. Over the search area, a grid is constructed with 100×100 cells. At the beginning of a test, 10 stationary and 10 mobile object targets are randomly placed in the search area. Positions of object targets are unknown to USVs, and mobile object targets will randomly alter their positions in every 10 second. A test is considered as complete if the search time exceeds the predefined threshold, (i.e., 500 s).

It should be mentioned that object targets employed in these ocean tests are virtual, i.e., their positions are labeled in the ground station (for the monitoring purpose), but no actual object targets are installed in the search area. This is because installing a large number of real object targets over a broad ocean region is difficult and expensive, especially for the mobile object targets. Besides, this experiment concentrates on evaluating the performance of the proposed method in organizing USV behaviors rather than the performance of the object detection sensor. Previous water test results have verified the capability of the onboard object detection sensor, i.e., the ability to capture object targets and produce object target positions (see Section 4.2). Thus, the image processing procedures are ignored in this group of ocean tests. An object target is considered captured if it is within the perception range of a USV and its cell confidence exceeds the threshold, i.e., 0.5.

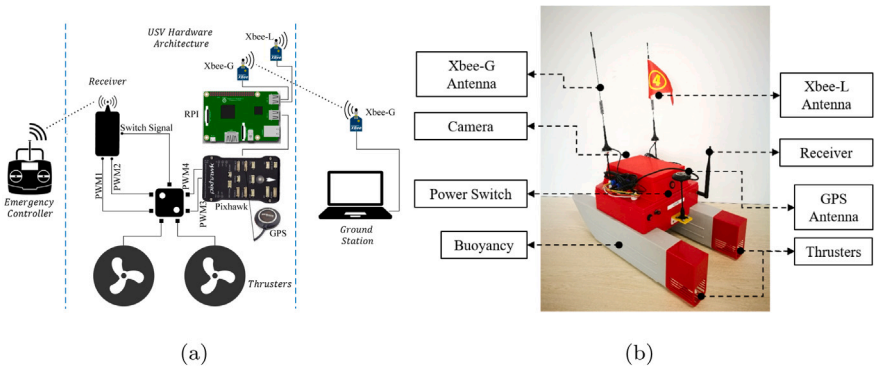


Fig. 11. (a) The hardware architecture of the USV, where RPI represents the onboard computer, i.e., Raspberry pi 3. (b) The configuration of components on a USV.

Fig. 12. The operated ocean region of the USV coordination algorithm, where (a) USVs form a formation to perform the search and (b) USVs cooperatively search for targets with the proposed coordination algorithm.

6.2.2. Results of formation control and uncoordinated algorithms

Fig. 13(a) presents USV search trajectories with the conventional formation control. The USV fleet starts at the bottom left corner of the search area and maintains a line formation with 5 m gap between two adjacent USVs. At $t = 500$ s, the fleet scans the entire search area in a lawn-mower pattern. Fig. 13(b) shows the grid confidence map from a USV generated based on the formation control algorithm. It can be observed that the fleet's knowledge about the search area is biased: the right half of the search area has higher cell confidence values compared to the left half (i.e., the right half of the grid confidence map is brighter than the left half). This is because the USV fleet starts the search from the left side and never revisits the detected areas. If a mobile object target moves to the left half of the search area, the USV fleet cannot detect this object target with the formation control algorithm. Thus, the search performance of this formation control algorithm is not ideal.

The search performance is also not desired for the uncoordinated algorithm. In this method, individual USVs are controlled by the proposed MDP without exchanging information among fleet members. Figs. 13(c) and 13(d) present the search trajectories of the USV fleet and the grid confidence map fetched from a USV at the end of the search, i.e., $t = 500$ s. It can be seen that the USV fleet cannot search the entire ocean region given the same amount of time as the formation control strategy. Besides, due to the lack of sharing information among the fleet members, the search efficiency is deteriorated since USVs repeatedly visit areas where other USVs have already searched. At the end of the search, most of the scanned areas are congregated near the bottom left corner of the search area as presented in Fig. 13(d). Snapshots in these two tests are presented in Fig. 14

6.2.3. Results of the proposed coordination algorithm

Fig. 15 demonstrates search results of the proposed coordination algorithm, where Fig. 15(a)–(d) present USV search trajectories at different time steps and Fig. 15(e)–(h) show the corresponding grid confidence maps.

Similar to previous tests, the USV fleet starts at the bottom left corner of the search area. At the beginning phase of the search, i.e., $t = 12$ s, USVs are scattered to explore the search area. When $t = 66$ s, it can be observed in Fig. 15(f) that regions in the left half of the search area have been explored. At $t = 116$ s, the USV fleet continues to search object targets at the right side of the search area. As presented in Fig. 15(g), the explored area almost covers the entire search region, including the remote regions in the top right corner. At $t = 500$ s, the USV fleet completes the search, and almost every region in the search area maintains a very high cell confidence value (i.e., > 0.6), indicating that the USV fleet knows the search area well. Once an object target is within the perception area of a USV, its location can be quickly detected.

Comparing the grid confidence map generated by three coordination algorithms at the end of the search, i.e., Figs. 13(b), 13(d), and 15(h), it can be observed that the USV fleet with the proposed coordination algorithm maintains high cell confidence values over the entire search area, which is beneficial for timely detecting object targets. This result verifies the validity of the proposed coordination algorithm. Snapshots of this test are presented in Fig. 16. Although the sea condition presented in photos is calm, we expect that the search performance of the proposed method will not be affected significantly in severe sea conditions. This is because the proposed method considers the environmental disturbance (waves and wind) in the path planning procedure, where a probabilistic state transition process is defined to calculate USV future locations. It should be mentioned that the actual search performance of the proposed method in severe sea conditions is also one of our future research directions.

6.3. Statistic analysis and discussion

To mathematically analyze results of the proposed coordination algorithm, the effective coverage rate and the number of detected object targets are calculated and compared with formation control and uncoordinated algorithms.

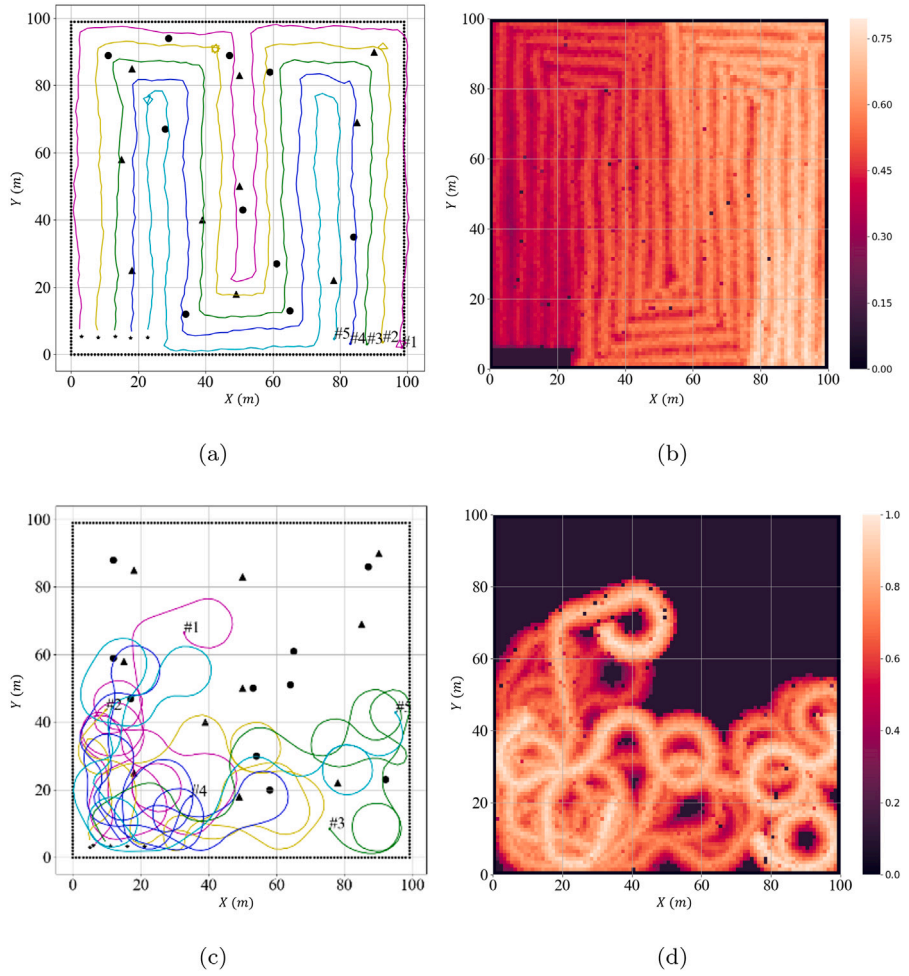


Fig. 13. Ocean test results with the formation control and uncoordinated algorithms. (a) USV search trajectories with the formation control algorithm. Large solid circles represent mobile object targets, and triangles indicate the stationary object targets. (b) The grid confidence map generated based on the search trajectories of the formation control algorithm. In particular, colors indicate confidence values: brighter areas indicate higher values of cell confidence. For the uncoordinated algorithm, search trajectories and the corresponding grid confidence map are presented in (c) and (d), respectively.

6.3.1. *Effective coverage rate*

The effective coverage rate (i.e., η) measures the efficiency of the implemented coordination algorithm for scanning the search area, which is defined as:

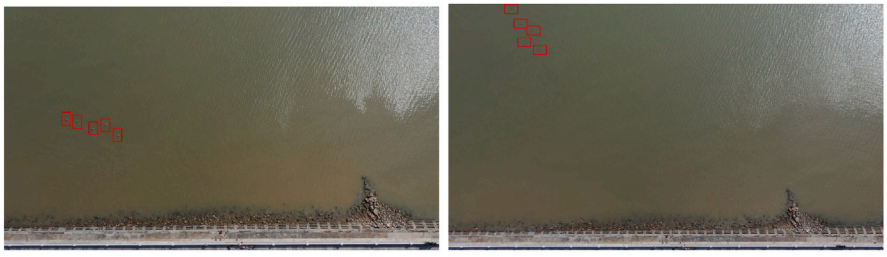
$$\eta = \frac{N_{\text{cov}}}{N_{\text{area}}}, \tag{17}$$

where N_{cov} is the number of effective searched cells and N_{area} is the number of cells in the search area. The term, effective searched cell, denotes that the cell confidence value of a cell exceeds the threshold, i.e., 0.5. The value of η reflects the effectiveness and efficiency of the implemented coordination algorithm: the higher the value, the better the search performance.

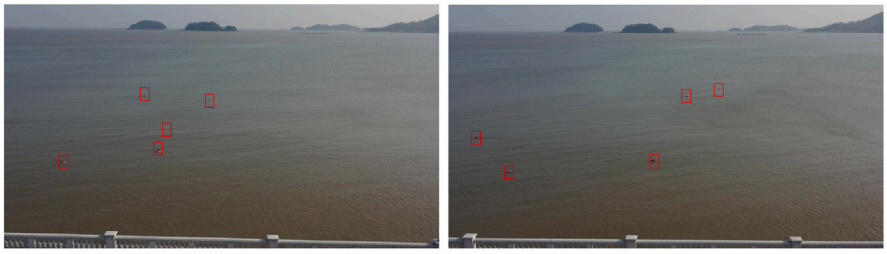
Fig. 17 presents plots of η calculated based on results of three coordination algorithms. For the proposed method, the value of η grows rapidly during the early phase of the search (i.e., from $t = 0$ s to $t = 350$ s), surpassing the other two methods. This is because with the proposed coordination algorithm, USVs can scatter to broadly explore the search area without the constraint of maintaining a formation pattern. Besides, by exchanging local sensing information with other fleet members, individual USVs can cooperatively search the environment and avoid visiting the already searched areas repeatedly. In the later phase of the search (i.e., from $t = 380$ s to $t = 500$ s), the increase rate of η drops due to the decrease of undetected areas. At the end of the search, i.e., $t = 500$ s, the value of η converges to 81%.

For the formation control strategy, the value of η increases linearly over the entire excursion since the pace of exploring the search area is fixed with the formation control strategy. When the search is complete at $t = 500$ s, the value of η reaches 84.3%, slightly larger than the proposed method. This is because the formation control strategy can thoroughly scan the search area with the lawn-mower trajectory pattern, while the proposed method focuses more on perceiving the environment as efficient as possible rather than scanning the entire search area. As for the uncoordinated algorithm, the value of η is less than 60% at the end of the search, resulting in the worst coverage rate among three coordination algorithms.

The high value of η leads to the better search performance of detecting object targets. Fig. 18 shows the number of detected object targets (Fig. 18(a) for stationary and Fig. 18(b) for mobile object targets) with three USV coordination algorithms over the search excursion. During the period from $t = 0$ s to $t = 350$ s, the proposed method finds the highest number of object targets (8 stationary targets and 8 mobile targets) thanks to the rapid growth of η. When $t = 500$ s, the formation control strategy finds 10 stationary object targets compared to 8 and 4 for the proposed and uncoordinated methods, respectively. This result verifies that the formation control strategy is suitable for detecting stationary object targets thanks to the ability of thoroughly scanning the search area. However, this algorithm is not efficient for detecting mobile object targets: at $t = 500$ s, the formation control strategy only finds 6 mobile object targets, while the proposed method finds the

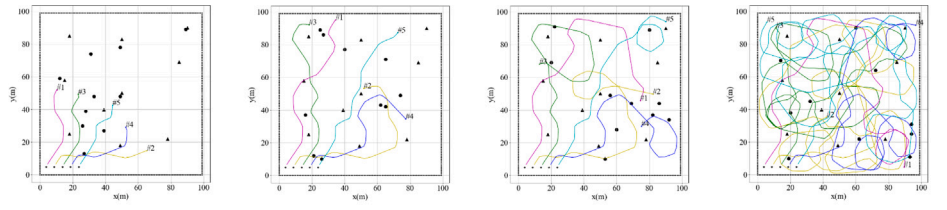


(a) Formation $t = 20$ s (b) Formation $t = 50$ s

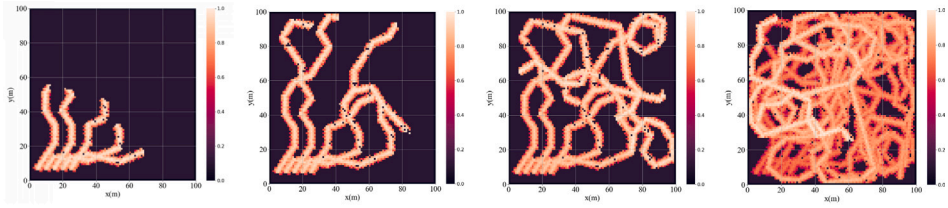


(c) Uncoordinated $t = 100$ s (d) Uncoordinated $t = 250$ s

Fig. 14. Snapshots of ocean tests with formation and uncoordinated control algorithms. (a) and (b) are snapshots in the formation control algorithm at $t = 20$ s and $t = 50$ s, respectively. (c) and (d) are snapshots in the uncoordinated control algorithm at $t = 100$ s and $t = 250$ s, respectively. The USVs' positions are highlighted with red squares in snapshots.

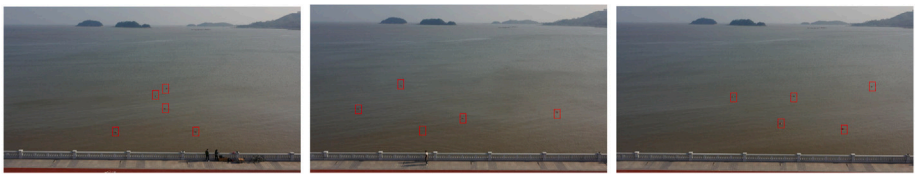


(a) $t = 12$ s (b) $t = 66$ s (c) $t = 116$ s (d) $t = 500$ s



(e) $t = 12$ s (f) $t = 66$ s (g) $t = 116$ s (h) $t = 500$ s

Fig. 15. Ocean test results with the proposed USV coordination algorithm. The first row of diagrams, i.e., (a)–(d), shows the USV search trajectories at different time steps. In these diagrams, large solid circles represent the positions of mobile object targets, and triangles indicate the stationary object targets' positions. The second row of graphs, i.e., (e)–(h), presents the grid confidence map at corresponding time steps with the trajectory diagrams. Confidence values are indicated with the darkness of colors: the higher the cell confidence values, the brighter the area will be. Black dots over the USV trajectories are generated due to the transmission failure, i.e., the USV does not receive the grid confidence value transmitted by other USVs.



(a) Coordination $t = 100$ s (b) Coordination $t = 250$ s (c) Coordination $t = 400$ s

Fig. 16. Snapshots of ocean tests with the proposed coordination algorithm. USVs' positions are highlighted with red squares.

Table 2
The number of detected stationary and mobile object targets for three coordination algorithms in repeat tests.

	Formation control		Uncoordinated method		The proposed method	
	Stationary object target	Mobile object target	Stationary object target	Mobile object target	Stationary object target	Mobile object target
Test 1	10	6	5	5	9	9
Test 2	10	5	4	6	8	9
Test 3	9	5	7	4	9	7
Test 4	10	6	6	6	10	7
Test 5	10	5	6	5	7	9
Test 6	10	7	5	6	8	8
Test 7	10	5	4	6	8	7
Test 8	10	6	6	6	10	8
Test 9	10	5	5	4	9	8
Test 10	9	5	8	5	8	8

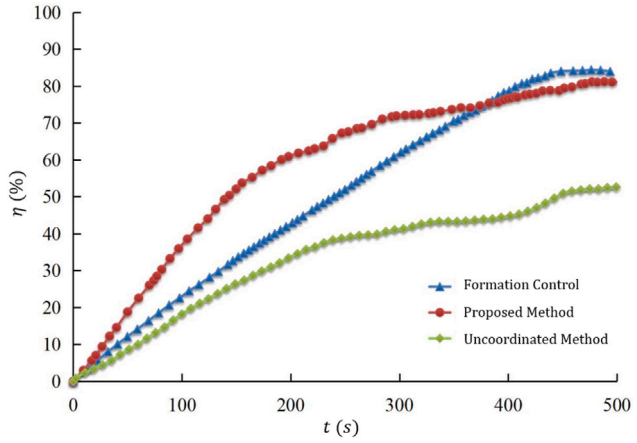


Fig. 17. The plot of effective coverage rates generated from results of three implemented coordination algorithms.

Table 3
Statistical results of repeat tests for three coordination algorithms.

	Total number of detected stationary targets	Averaged number of detected stationary targets	Total number of detected mobile targets	Averaged number of detected mobile targets
Formation control	98	9.8	55	5.5
Uncoordinated method	56	5.6	54	5.4
The proposed method	86	8.6	80	8

most mobile object targets (i.e., 8) compared to 5 for the uncoordinated method.

Results from this section reveal that the proposed coordination algorithm can quickly perceive the environment at the beginning of the search and reaches a comparable coverage rate with the formation control strategy at the end of the search, which is preferable for detecting both stationary and mobile object targets to achieve the desired search efficiency.

6.3.2. Repeat tests

In this group of tests, three USV coordination algorithms, i.e., the formation control, uncoordinated, and proposed coordination algorithms, are repeatedly performed 10 times in the aforementioned ocean test environment to evaluate the ability of finding object targets.

Table 2 lists the number of detected object targets with three coordination algorithms in every test, and Table 3 shows the statistical results of all tests. It can be observed in Table 2 that the proposed method achieves a comparable search performance with the formation control strategy in detecting stationary object targets. For detecting mobile object targets, the proposed algorithm achieves the best performance, while the number of detected mobile object targets surpasses other two methods in every test.

From Table 3, the total number of detected stationary object targets is 98 for the formation control strategy (the averaged value is 9.8), while this number is 56 for the uncoordinated method and 86 for the proposed method. The total number of detected mobile object target is 80 for the proposed method, which significantly outperforms the formation control strategy (55) and the uncoordinated method (54). It should be mentioned that detecting mobile object targets is more significant than detecting the stationary counterpart for practical applications since floating objects, such as floating survivors, are not static in the ocean environment.

To the best of authors' knowledge, our method is the first RL-based swarm coordination algorithm for a search-and-rescue using USVs. The state-of-art RL-based coordination algorithms for USVs include deep deterministic policy gradient (DDPG)-based (Woo et al., 2019) and deep Q network (DQN)-based methods (Jin et al., 2019), but both methods were designed to control a USV fleet to follow a pre-define path. In our problem, the search path is unknown, and the coordination algorithm should intelligently calculate a search path that achieves the pre-defined goals. In the future, we will compare our method with state-of-art if there are similar RL-based swarm coordination methods for search-and-rescue tasks using USVs.

In general, compared to the conventional formation control and uncoordinated algorithms, ocean test results indicate that the proposed algorithm is more desirable for searching object targets. Given the same amount of time, USVs with the proposed algorithm can intelligently scan the search area and detect object targets more efficiently and cooperatively. Besides, without the requirement to maintain a formation, USV control efforts are reduced (i.e., USVs do not need to adjust gestures and positions to maintain the formation), which is beneficial for saving onboard power and supporting the USV fleet to search broader water regions.

7. Conclusion

This article presents a coordination algorithm for organizing a fleet of USVs to search object targets in an unknown water region. The major steps are twofold: modeling and planning. In the modeling procedure, a grid confidence map is constructed over the gridded search area, which indicates how much information the fleet knows about the search area. This information refers to whether a region contains object targets. During the fleet maneuver, the grid confidence map is updated via local observations. In the planning procedure, an MDP is employed to model the search behaviors of a single USV, where reward functions are defined based on the grid confidence map. With the information provided by the grid confidence map, a USV is encouraged to explore new areas and prevented to search the already detected regions repeatedly. The policy iteration algorithm is adapted to solve for the optimal

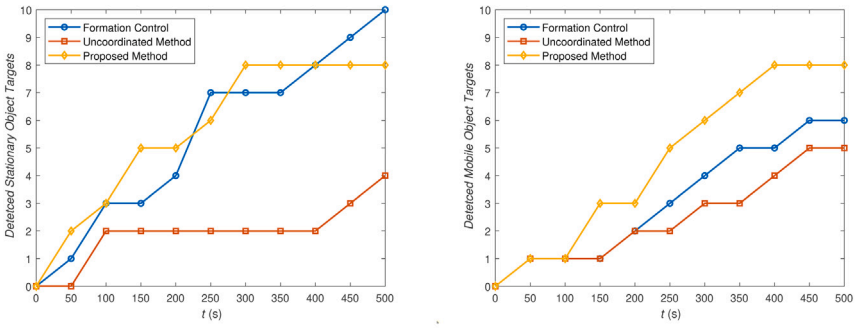


Fig. 18. (a) The number of detected stationary object targets. (b) The number of detected mobile object targets.

policy, i.e., search routes. With action policy constraints, the inter-vehicle collisions are prevented. The validity of the proposed algorithm is verified by implementing it in ocean tests. Compared to conventional formation control algorithm and uncoordinated algorithm, ocean test results demonstrate that the proposed method is preferable for searching object targets in the ocean environment.

CRediT authorship contribution statement

Runlong Miao: Conceptualization, Methodology, Software, Validation. **Lingxiao Wang:** Formal analysis, Methodology, Validation, Writing – original draft, Writing – review & editing. **Shuo Pang:** Supervision, Methodology, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Almeida, J., Silvestre, C., Pascoal, A., 2010. Cooperative control of multiple surface vessels in the presence of ocean currents and parametric model uncertainty. Internat. J. Robust Nonlinear Control 20 (14), 1549–1565.

Caccia, M., Bono, R., Bruzzone, G., Spirandelli, E., Veruggio, G., Stortini, A., Capodaglio, G., 2005. Sampling sea surfaces with SESAMO: an autonomous craft for the study of sea-air interactions. IEEE Robot. Autom. Mag. 12 (3), 95–105.

Chen, Y.Q., Wang, Z., 2005. Formation control: a review and a new consideration. In: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, pp. 3181–3186.

Do, K.D., 2011. Formation control of underactuated ships with elliptical shape approximation and limited communication ranges.

Glotzbach, T., Schneider, M., Otto, P., 2015. Cooperative line of sight target tracking for heterogeneous unmanned marine vehicle teams: From theory to practice. Robot. Auton. Syst. 67, 53–60.

Guo, X., Ji, M., Zhao, Z., Wen, D., Zhang, W., 2020. Global path planning and multi-objective path control for unmanned surface vehicle based on modified particle swarm optimization (PSO) algorithm. Ocean Eng. 216, 107693.

Jin, K., Wang, H., Yi, H., et al., 2019. End-to-end trajectory tracking algorithm for unmanned surface vehicle using reinforcement learning. In: The 29th International Ocean and Polar Engineering Conference. International Society of Offshore and Polar Engineers.

Li, T., Zhao, R., Chen, C.P., Fang, L., Liu, C., 2018. Finite-time formation control of under-actuated ships using nonlinear sliding mode control. IEEE Trans. Cybern. 48 (11), 3243–3253.

Liu, Y., Bucknall, R., 2016. The angle guidance path planning algorithms for unmanned surface vehicle formations by using the fast marching method. Appl. Ocean Res. 59, 327–344.

Liu, Y., Bucknall, R., Zhang, X., 2017. The fast marching method based intelligent navigation of an unmanned surface vehicle. Ocean Eng. 142, 363–376.

Liu, Y., Peng, Y., Wang, M., Xie, J., Zhou, R., 2020. Multi-USV system cooperative underwater target search based on reinforcement learning and probability map. Math. Probl. Eng. 2020.

Liu, Z.-Q., Wang, Y.-L., Wang, T.-B., 2018. Incremental predictive control-based output consensus of networked unmanned surface vehicle formation systems. Inform. Sci. 457, 166–181.

Meyer, E., Heiberg, A., Rasheed, A., San, O., 2020. Colreg-compliant collision avoidance for unmanned surface vehicle using deep reinforcement learning. IEEE Access 8, 165344–165364.

Miao, R., Pang, S., Jiang, D., 2019. Development of an inexpensive decentralized autonomous aquatic craft swarm system for ocean exploration. J. Mar. Sci. Appl. 18 (3), 343–352.

Naeem, W., Xu, T., Sutton, R., Tiano, A., 2008. The design of a navigation, guidance, and control system for an unmanned surface vehicle for environmental monitoring. Proc. Inst. Mech. Eng. Part M 222 (2), 67–79.

Peng, Z., Jiang, Y., Wang, J., 2020. Event-triggered dynamic surface control of an underactuated autonomous surface vehicle for target enclosing. IEEE Trans. Ind. Electron. 68 (4), 3402–3412.

Peng, Z., Wang, D., Chen, Z., Hu, X., Lan, W., 2012. Adaptive dynamic surface control for formations of autonomous surface vehicles with uncertain dynamics. IEEE Trans. Control Syst. Technol. 21 (2), 513–520.

Peng, Z., Wang, D., Wang, J., 2015. Cooperative dynamic positioning of multiple marine offshore vessels: A modular design. IEEE/ASME Trans. Mechatronics 21 (3), 1210–1221.

Peng, Z., Wang, J., Wang, D., 2017. Distributed maneuvering of autonomous surface vehicles based on neurodynamic optimization and fuzzy approximation. IEEE Trans. Control Syst. Technol. 26 (3), 1083–1090.

Qin, Z., Lin, Z., Yang, D., Li, P., 2017. A task-based hierarchical control strategy for autonomous motion of an unmanned surface vehicle swarm. Appl. Ocean Res. 65, 251–261.

Redmon, J., Farhadi, A., 2017. YOLO9000: better, faster, stronger, In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263–7271.

Roberts, G.N., Sutton, R., 2006. Advances in Unmanned Marine Vehicles, Vol. 69. Iet.

Shannon, C.E., 2001. A mathematical theory of communication. ACM SIGMOBILE Mobile Comput. Commun. Rev. 5 (1), 3–55.

Shojaei, K., 2015. Leader–follower formation control of underactuated autonomous marine surface vehicles with limited torque. Ocean Eng. 105, 196–205.

Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al., 2016. Mastering the game of go with deep neural networks and tree search. Nature 529 (7587), 484.

Sun, X., Wang, G., Fan, Y., Mu, D., Qiu, B., 2020. A formation autonomous navigation system for unmanned surface vehicles with distributed control strategy. IEEE Trans. Intell. Transp. Syst..

Sutton, R.S., Barto, A.G., 2018. Reinforcement Learning: An Introduction. MIT Press.

Tan, Y., Zheng, Z., 2013. Research advance in swarm robotics. Defence Technol. 9 (1), 18–39.

Tan, G., Zhuang, J., Zou, J., Wan, L., Sun, Z., 2020a. Artificial potential field-based swarm finding of the unmanned surface vehicles in the dynamic ocean environment. Int. J. Adv. Robot. Syst. 17 (3), 1729881420925309.

Tan, G., Zou, J., Zhuang, J., Wan, L., Sun, H., Sun, Z., 2020b. Fast marching square method based intelligent navigation of the unmanned surface vehicle swarm in restricted waters. Appl. Ocean Res. 95, 102018.

Wang, Y., Han, Q., Fei, M., Peng, C., 2018. Network-based T–S fuzzy dynamic positioning controller design for unmanned marine vehicles. IEEE Trans. Cybern. 48 (9), 2750–2763.

Woo, J., Yu, C., Kim, N., 2019. Deep reinforcement learning-based controller for path following of an unmanned surface vehicle. Ocean Eng. 183, 155–166.

Xia, G., Han, Z., Zhao, B., Wang, X., 2020. Local path planning for unmanned surface vehicle collision avoidance based on modified quantum particle swarm optimization. Complexity 2020.

Xin, J., Li, S., Sheng, J., Zhang, Y., Cui, Y., 2019. Application of improved particle swarm optimization for navigation of unmanned surface vehicles. Sensors 19 (14), 3096.

Yang, Z., Wang, Y., Liu, Z., 2014. Sliding mode robust control for formation of multiple underactuated surface vessels. In: Proceeding of the 11th World Congress on Intelligent Control and Automation. IEEE, pp. 3775–3780.

Yin, S., Yang, H., Kaynak, O., 2016. Coordination task triggered formation control algorithm for multiple marine vessels. IEEE Trans. Ind. Electron. 64 (6), 4984–4993.

Zhao, Y., Qi, X., Ma, Y., Li, Z., Malekian, R., Sotelo, M.A., 2020. Path following optimization for an underactuated USv using smoothly-convergent deep reinforcement learning. IEEE Trans. Intell. Transp. Syst..

Zhou, X., Wu, P., Zhang, H., Guo, W., Liu, Y., 2019. Learn to navigate: cooperative path planning for unmanned surface vehicles using deep reinforcement learning. IEEE Access 7, 165262–165278.