

Online paths planning method for unmanned surface vehicles based on rapidly exploring random tree and a cooperative potential field

*International Journal of Advanced
Robotic Systems*

March–April 2022: 1–22


© The Author(s) 2022

Article reuse guidelines:

sagepub.com/journals-permissions

DOI: 10.1177/117298806221089777

journals.sagepub.com/home/arx

Naifeng Wen^{1,2} , Lingling Zhao³, Ru-Bo Zhang^{1,2} , Shuai Wang⁴,
Guanqun Liu^{1,2}, Junwei Wu^{1,2} and Liyuan Wang¹

Abstract

The unstructured, dynamic marine environmental information and the cooperative obstacle avoidance problem greatly challenge the online path planner for unmanned surface vehicles. Efficiency and optimization are crucial for online path planning schemes. Thus, we proposed an algorithmic combination of the optimal rapidly exploring random tree and artificial potential field methods. First, we built a repulsive potential field by considering the relative velocity and position of the unmanned surface vehicle to obstacles and the international regulations for preventing collisions at sea, wherein we designed a repulsive force calculation method using radar readings to avoid irregular obstacles. Then, we guided the sampling process of rapidly exploring random tree using the potential field to accelerate the convergence rate of rapidly exploring random tree to low-cost obstacle avoidance paths. Finally, we planned for multiple paths based on the leader–follower architecture with the guidance of a cooperative potential field. In the experiments, the proposed method consistently outperformed the benchmark methods. We also verified the effectiveness of the algorithmic modifications by conducting ablation experiments.

Keywords

Optimal rapidly exploring random tree, artificial potential field, online path planning, international regulations for preventing collisions at sea, cooperative collision avoiding

Date received: 13 April 2021; accepted: 08 March 2022

Topic: Mobile Robots and Multi-Robot Systems

Topic Editor: Nak-Young Chong

Associate Editor: Changjoo Nam

Introduction

Online paths planning (OPP) is necessary for cooperative tasks of autonomic unmanned surface vehicles (USVs), such as cooperative search and rescue in the open sea with partially unknown locations and shapes of static obstacles (e.g. islands) and dynamic obstacles (DOs; e.g. ships) that unpredictably appear, disappear, or move.¹ The manipulators of the planned paths are assumed to be small, lightweight USVs with high maneuverability and relatively low speed. Thus, USVs can follow the path to avoid complex obstacles. The simplified state vector of USVs comprises

¹School of Mechanical and Electronic Engineering, Dalian Minzu University, Dalian, China

²Key Laboratory of Intelligent Perception and Advanced Control, Dalian Minzu University, Dalian, China

³Faculty of Computing, Harbin Institute of Technology, Harbin, China

⁴College of Mathematics and Informatics, Digital Fujian Internet-of-Things Laboratory of Environmental Monitoring, Fujian Normal University, Fujian, China

Corresponding author:

Ru-Bo Zhang, College of Mechanical and Electronic Engineering, Dalian Minzu University, Liaohe west 18, Economic and Technological Development Zone, Dalian, Liaoning 116600, China.

Email: zhangrubo@dlmu.edu.cn



Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (<https://creativecommons.org/licenses/by/4.0/>) which permits any use, reproduction and distribution of the work without

further permission provided the original work is attributed as specified on the SAGE and Open Access pages (<https://us.sagepub.com/en-us/nam/open-access-at-sage>).

surge, sway, and yaw variables. The planning space is projected from the state space.

Efficiency is the first requirement of the OPP method for the quick response to real-time sensed data to capture cooperative paths. However, the path planning problem in dynamic environments is computationally complex. Meanwhile, cooperative constraints, such as the position coupling between USVs in the obstacle avoidance (OA) process,²⁻⁴ greatly challenge the online path planner. If paths are planned individually, the computational complexity is probably unacceptable. Moreover, the high path quality is also a crucial performance indicator for the online planner. Besides the geometric features of paths, the planner should also consider the influence of time-varying sea current.⁵⁻⁷ The studies of literature⁷⁻⁹ suggested that the energy-efficient path should contain various downstream sections. In summary, the OPP technology for USVs is far from mature. This observation is the motivation of this study.

The optimal OA control for a single USV is complicated because of the coupled differential equations with boundary conditions and motion constraints.¹⁰⁻¹⁵ Thus, the problem is decoupled into subproblems,¹⁶ such as path planning and optimal control. The model predictive control (MPC) frame is a typical decoupling method that requires the online path planner to provide committed local paths iteratively in short periods. On the basis of MPC, we focus on the path planning problem, leaving the complicated optimal control problem to the path executive module. We initially plan a local path reaching a subgoal that is selected within the sensing domain of USVs. Another local path planning iteration starts when USVs execute the available path. The iterations roll forward until the USVs achieve the goal.

The intelligent searching and reasoning methods are often too computationally complex to apply online.¹⁷ The prevalent reinforcement learning method¹⁸ has difficulty in solving the multiple paths planning problems. The artificial potential field (APF) method quickly queries paths in the gradient descent direction of the potential field that attracts the USV to the target and repulses the USV away from obstacles.

To address the problem of the goal nonreachable with obstacles nearby, Ge and Cui improved APF by defining a repulsive potential field taking positions of the USV and the goal into account.^{19,20} Repulsive force was divided into components in the direction from the obstacle to the USV and in the vertical direction of the relative velocity of the USV with respect to the obstacle in the study.²⁰ The studies of literature^{21,22} promoted the potential field definition by considering more OA parameters.

Ge et al. employed an instant-goal-driven method to compute the repulsive force using radar readings.²³ However, the method did not consider the situation when radar readings are from multiple obstacles. Solari et al. calculated the point potential at a certain distance from the obstacle on each beam of the mechanical scanning sonar and

selected the motion direction of the vehicle by the lowest potential.²⁴

Lyu and Yin et al. improved the repulsive field definition by considering OA angle and velocity.^{25,26} Wang et al. proposed a ring-shaped repulsive potential field for USV swarm control.¹⁵ Wang et al. developed APF by combining it with the ship domain model to consider the DO's velocity and course.²⁷ To bypass obstacles smoothly, Qin et al. added a velocity damping coefficient to the repulsive potential function.²⁸

Balch and Hybinette employed virtual forces toward the attachment sites of USVs to keep the formation²⁹ and investigated the social potential for scalable vehicle formation.³⁰ Ge and Fua investigated queues and artificial potential trenches for multirobot formation.³¹ Sun et al. studied the collision avoidance problem for UAVs using APF.³²

However, APF depends on the elegant mathematical analysis based on known, static environmental information, which is unpractical. Uncertainties in the environmental information and the USV motion model make the potentials incomparable, even causing the local optimum. Rapidly exploring random tree (RRT)-based methods were proposed to plan/replan feasible paths quickly with differential, nonholonomic, and inequality constraints.³³⁻³⁵ The optimal rapidly exploring random tree (RRT*) asymptotically converges to the optimal path by incrementally rewiring the path tree. After new states are added to the path tree, they are also considered as replacement parents for existing nearby states in the path tree. RRT* can conveniently solve the local optimum.³³⁻³⁵ However, the efficiency of RRT* is possibly low.^{36,37}

When an initial solution path is found, the redundant points of the path are pruned by the RRT*-Smart³⁶ method, and the remaining states as used as biases for further sampling. However, the method may reduce the probability of finding a different homotopy class of the initial solution and further violate the RRT* assumption of uniform density.³⁷

Gammell et al.³⁷ presented the informed RRT* method to improve the efficiency of RRT* by focusing the path planning in terms of directly sampling a prolate hyperspheroid subset that is defined as $x_{ellipse} = Lx_{ball} + x_{center}$, where x_{center} is the center of the hyper-ellipsoid in terms of its two f_{local} points that are often set as the start and goal points, x_{ball} is the unit ball, $L = \text{diag}\{c_{best}/2, 0.5*(c_{best}^2 - c_{min}^2)^{1/2}, \dots, 0.5*(c_{best}^2 - c_{min}^2)^{1/2}\}$, c_{best} is the cost of the current best path, c_{min} is the distance between the start and goal points, and the dimension of the diagonal matrix L equals that of the planning space.

Otte and Emilio³⁸ proposed the RRT^X method for efficiently avoiding DOs. Rewiring operations cascade down the affected branches using existing nodes to remodel the existing search graph and repair the shortest path-to-goal subtree. Replanning efficiency is improved by maintaining graph connectivity information in local neighbor sets stored at each node with the expected number of $O(\log n)$, where

n is the number of nodes, as well as aborting rewiring cascades once the graph becomes consistent. Moreover, graph consistency ensures that the newly added nodes and edges are useful for improving the solution.

The study by Wang et al.³⁹ improved the probabilistic sampling with the 2D Gaussian mixture model to sample around obstacles for short OA paths. The study in Jaillot et al.⁴⁰ used the cached historical paths as heuristics to improve replanning efficiency. Transition-based RRT promoted sampling in terms of the environment configuration-space cost maps.⁴¹

The above methods only consider the DO to be errant. Wen et al.⁴² assumed that USV can negotiate with DOs and then sampling is biased by the International Regulations for Preventing Collisions at Sea (COLREGs).

As one of the state-of-the-art OPP solutions, the potential field was combined with RRT*.^{41–44} Qureshi et al. directionalized random samples of RRT* through potential functions for reducing the number of planning iterations.^{43–45} Wang et al. solved the motion planning problem by the bidirectional potential guided RRT* method.⁴⁶

In this article, the RRT* efficiency is improved by the following schemes: the local potential field guides the planning toward reasonable areas in terms of the real-time feedback of sensors; the potential field is improved by considering the velocities of USV and DO; and the given environmental information is leveraged by selecting subgoals referring to the offline planned global path. First, the virtual leader's local path is planned to obtain multiple cooperative USV paths. Then, the expected locations of USVs referring to the leader are defined, and multiple paths are planned by the distributed local planners in each USV. The cost distance for the RRT* rewiring is defined by considering sea current, path length, and path turns to reduce the energy expenditure of paths.

Therefore, the main contributions of this study are two-fold. (1) We improve the efficiency of RRT* to accelerate the convergence to energy-efficient and cooperative OA paths by the guidance of a newly built potential field in dynamic environments, wherein the repulsive potential field considers the OA-related velocities and rules. It is calculated from the radar readings. (2) The online multi-path planning architecture is constructed based on the MPC and virtual leader frameworks.

Sketch of the paths planning system for USVs

The hierarchical architecture is divided into submodules,^{8,9,27,28} as shown in Figure 1. On the global path planning layer, the potential field used for capturing the global reference path is built in terms of the given environmental information, and the parameters are set empirically. During the online planning procedure, the parameters remain unchanged. Then, the OPP module plans for feasible, low-cost paths for the virtual leader, wherein samples of RRT* are directionalized by the potential field. On the cooperative path planning layer, the cooperation attachment sites of the USV formation are defined by following the virtual leader, and the cooperation potential field is then constructed. Finally, the OA paths are planned locally for USVs.

Objective of the OPP

Three important properties of the paths are considered during the OPP process, i.e. low collision probability, feasibility, and the likelihood of success.^{23,12} The likelihood of success requires paths to be energy efficient with few aggressive turns. The planning objective expression is listed in the following

$$\operatorname{argmin} \left\{ \sum_{k=1}^M \sum_{i=1}^N e_{i,k} \cdot v_{i,k} \cdot \operatorname{length}(\operatorname{PlanPath}(p_{i,k}, p_{i+1,k})) \right\}, \quad s.t. \quad F_{\text{cons}} = 0 \quad (1)$$

where $F_{\text{cons}} = 0$ means the feasible path needs to satisfy motion constraints, M is the number of USVs, N is the number of the discrete waypoints, $e_{i,k}$ denotes the energy cost against the sea current on a path section $\tau_{i,k}$ between adjacent waypoints: $p_{i,k}$ and $p_{i+1,k}$, of the k th USV, $v_{i,k}$ denotes the control cost on $\tau_{i,k}$, and $\operatorname{length}(\cdot)$ represents the length of $\tau_{i,k}$.

USV motion model

Usually, a physical USV model is simplified to establish a control model that satisfies system requirements, and the characteristics of the USV structure are moderately simplified.

We truncated the freedom degrees of USVs to surge, sway, and yaw for the OPP burden reduction. That is also because the influence of heave, pitch, and rolling motions is supposed to be relatively small on the OPP procedure in ordinary sea states. Figure 2 shows the Earth-fixed inertial frame $\{i\}$ and the body-fixed frame $\{b\}$. The positive X-axis of $\{b\}$ is in the USV heading direction, and the origin locates at the barycenter of USV. The vehicle is well designed to be symmetric so that the barycenter coincides with the center of the hull.

The experimental dynamic model is listed below according to the study in Mousazadeh and Kiapey⁴⁷

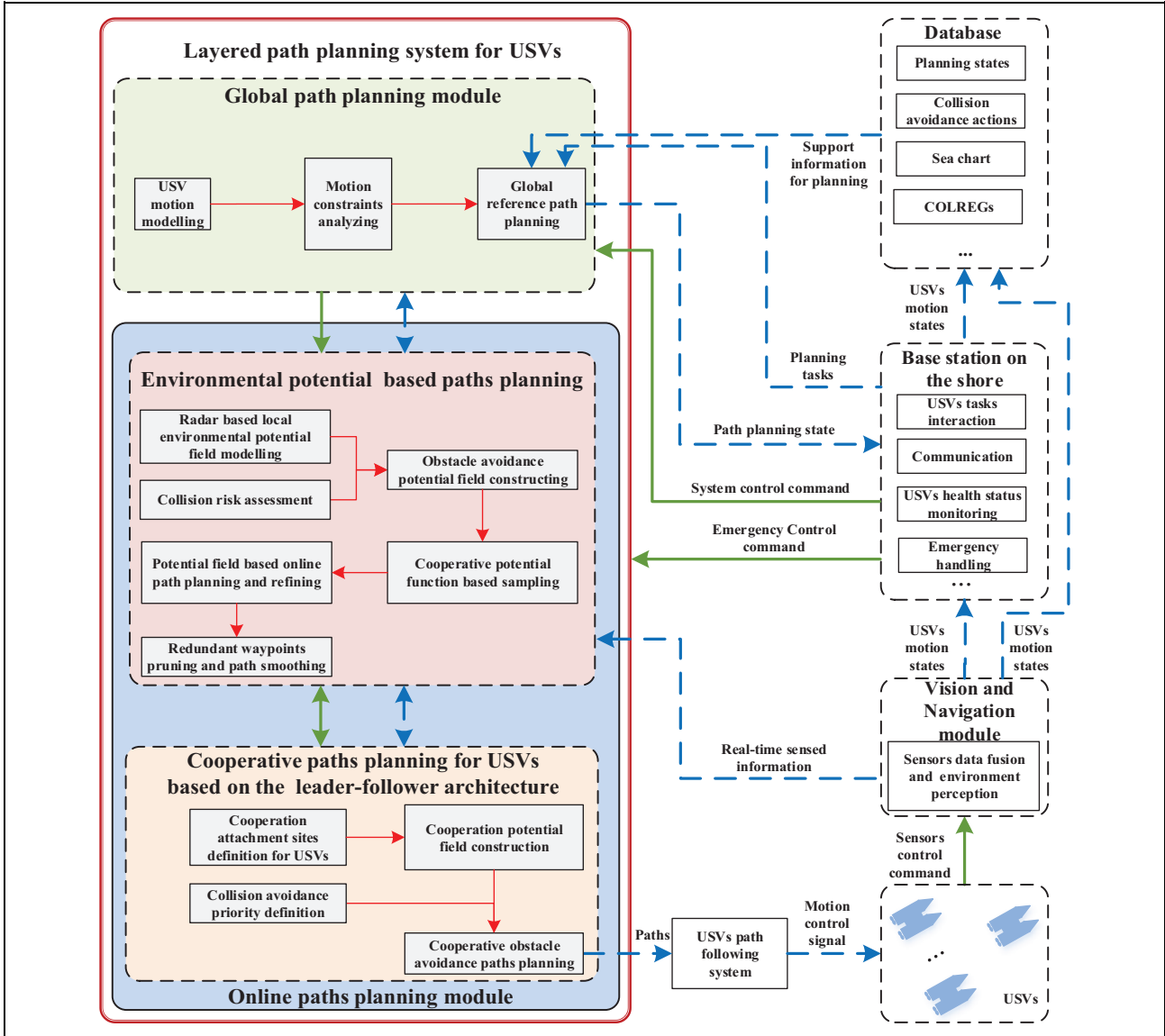


Figure 1. Hierarchically cooperative paths planning system for USVs. USV: unmanned surface vehicle.

$$\left\{ \begin{array}{l}
 \dot{x} = dx/dt = u \cdot \cos\theta - v \cdot \sin\theta \\
 \dot{y} = dy/dt = u \cdot \sin\theta + v \cdot \cos\theta \\
 \dot{\theta} = d\theta/dt = r \\
 \dot{u} = du/dt = \frac{m_{22}}{m_{11}} \cdot v \cdot r - \frac{d_u}{m_{11}} \cdot u - \sum_{i=2}^3 \frac{d_{ui}}{m_{11}} \cdot |u|^{i-1} \cdot u + \frac{1}{m_{11}} \cdot \tau_u \\
 \dot{v} = dv/dt = \frac{m_{11}}{m_{22}} \cdot u \cdot r - \frac{d_v}{m_{22}} \cdot v - \sum_{i=2}^3 \frac{d_{vi}}{m_{22}} \cdot |v|^{i-1} \cdot v \\
 \dot{r} = dr/dt = \frac{(m_{11} - m_{22})}{m_{33}} \cdot u \cdot v - \frac{d_r}{m_{33}} \cdot r - \sum_{i=2}^3 \frac{d_{ri}}{m_{33}} \cdot |r|^{i-1} \cdot r + \frac{1}{m_{33}} \cdot \tau_r
 \end{array} \right. \quad (2)$$

where $[\dot{x}, \dot{y}, \dot{\theta}]^T$ and $[u, v, r]^T$ are the velocity vectors in $\{i\}$ and $\{b\}$, respectively, and $d_u, d_v, d_r, d_{ui}, d_{vi},$ and d_{ri} ($i = 2, 3$) are the hydrodynamic damping weights. The m_{ii} ($i = 1, 2, 3$) denotes the mass parameters, and the mass matrix is diagonal. τ_u and τ_r denote the surge force and yaw moment, respectively. The parameters in Table 1 refer to a real USV, and they come from the study in Mousazadeh and Kiapey.⁴⁷ $d_{u2} = 0.2d_u, d_{v2} = 0.2d_v, d_{r2} = 0.2d_r, d_{u3} = 0.1d_u, d_{v3} = 0.1d_v, d_{r3} = 0.1d_r, L,$ and W are the length and width of the hull, respectively.

To guarantee the dynamic feasibility of the resulting path, we take motion constraints into account during the OPP process. The inequality constraints of the maximum velocity and acceleration are defined by the performance limitations of USV according to formula (2). The minimum turning radius is set as $r_{\min} = 4(u^2 + v^2) / \omega_{\max}$, where ω_{\max} is the maximum angular acceleration of USV, thus the extendable domain for a path tree node is fan-shaped. The nonholonomic differential of USV motion constraint is $\sin\theta \cdot dx - \cos\theta \cdot dy = 0$, where dx and dy are the X and Y increments in $\{i\}$, respectively. The constraints are checked during each path extension iteration to ensure the feasibility of the planned path sections.

Cooperative potential functions design

Collision detection

When a DO enters the working domain of USV, the collision cone scheme was improved to assess the collision situation by the intention of DO,¹⁰ and the collision navigational angle ranges were computed according to the relative velocity of the USV with respect to DO

$$\begin{cases} v_\theta = v_i \sin(\alpha - \theta) - v_o \sin(\beta - \theta) \\ v_r = v_i \cos(\alpha - \theta) - v_o \cos(\beta - \theta) \end{cases} \quad (3)$$

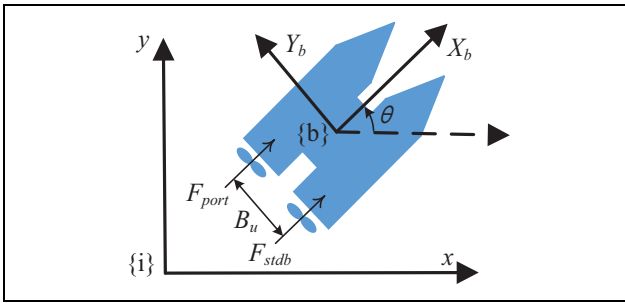


Figure 2. Schematic of USV planning frame system. USV: unmanned surface vehicle.

Table 1. Parameters of the USV dynamic model.

m_{11} (kg)	m_{22} (kg)	m_{33} (kg \times m ²)	d_u (kg/s)	d_v (kg/s)	d_r (kg \times m ² /s)	L (m)	W (m)
25.8	33.8	2.7	2	7	0.5	1.5	0.5

USV: unmanned surface vehicle.

Figure 3 illustrates that v_{io} is the relative velocity of the USV with respect to the DO, r_o is the safe radius from the USV to the DO, $p_i=(x_i, y_i)$ denotes the position of the USV, $p_o=(x_o, y_o)$ denotes the position of the DO, v_θ is the component of v_{io} on the direction perpendicular to the line $p_i p_o$, C is the intersection of the expected USV path on the safe circle centering at the DO if the USV and the DO maintain their velocities, and v_r is the component of v_{io} on the direction of $p_i p_o$. If $v_\theta=0$, then the relative trajectory of the USV to the obstacle is right on the line of $p_i p_o$. If $v_r>0$, then the projection of v_{io} on $p_i p_o$ is positive, and the USV has a tendency of becoming close to the obstacle; otherwise, the USV is more likely to separate from the DO. Figure 3 shows that collision probably occurs at point C . If $v_r>0$ and the direction of v_{io} is within the domain from $p_i A$ to $p_i B$ that are the tangents of the circular bounding box of the DO from the USV location, then collision possibly occurs. $(v_\theta)_{p_i A}$ and $(v_\theta)_{p_i B}$ are the components of v_{io} in the direction perpendicular to lines $p_i A$ and $p_i B$, respectively. Thus, the following collision detection condition can be obtained.

If one of the conditions, (1) $v_\theta=0$ and $v_r>0$; (2) $(v_\theta)_{p_i A} (v_\theta)_{p_i B} \leq 0$ and $v_r>0$, is met, then USV is probably collision with the obstacle. The proof of the second condition refers to in appendix 1. The practical style of the second condition is as: $d_{io}^2 v_\theta^2 \geq r_o^2 (v_r^2 + v_\theta^2)$, where d_{io} is the distance between USV and DO. The collision detection algorithm is as:
$$\begin{cases} v_\theta = 0 \text{ and } v_r > 0 \\ v_r > 0 \text{ and } d_{io}^2 \cdot v_\theta^2 \geq r_o^2 \cdot (v_r^2 + v_\theta^2) \end{cases}$$

Cooperative potential functions design for avoiding obstacles

The quadratically ring-shaped attractive potential field is constructed around the goal

$$U_{att}(q) = 0.5 \xi d_{it}^2 \quad (4)$$

where d_{it} is the distance from USV to the target. The attractive force is calculated by $F_{att}(q) = \xi d_{it} n_{it}$, where $n_{it} = -\nabla \rho(q_i, q_{target})$ is the direction unit vector from USV to the target, and we set ξ to be 1.

We defined a ring-shaped repulsive potential field around each obstacle at the region: $r_o < d_{io} < R_o$, where R_o is the radius to conduct the OA action. We set R_o to be 5 kilometers in open water areas and 500 m in narrow passage areas empirically. The repulsive force from obstacles is calculated as follows, and n is the number of obstacles

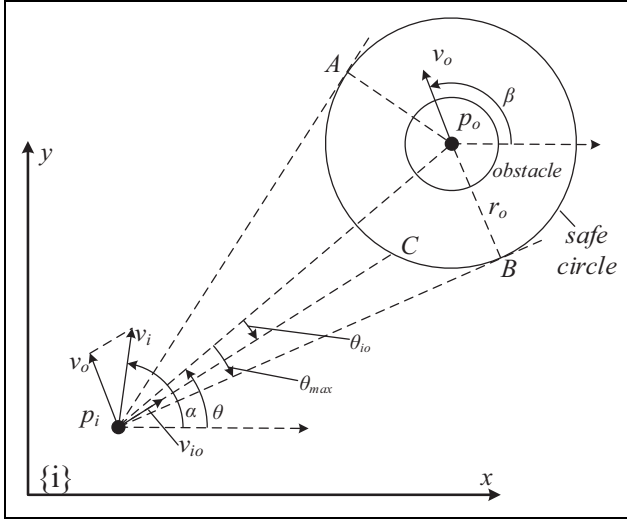


Figure 3. Illustration of the collision detection via the collision cone method.

$$\phi_{OA} = \sum_{i=1}^n \left(\frac{1}{2} \cdot \left(\frac{1}{d_{io}} - \frac{1}{r_o} \right)^2 \cdot d_{it}^2 \right) \quad (5)$$

$$F_{io} = \sum_{i=1}^n \left(\frac{\partial \phi_{OA}}{\partial v_{io}} \cdot \frac{\partial v_{io}}{\partial v} + \frac{\partial \phi_{OA}}{\partial d_{io}} \cdot \frac{\partial d_{io}}{\partial p} + \frac{\partial \phi_{OA}}{\partial d_{it}} \cdot \frac{\partial d_{it}}{\partial p} + \frac{\partial \phi_{OA}}{\partial v_{io}} \cdot \frac{\partial v_{io}}{\partial p} \right) \quad (6)$$

Because $v_{io} = (v_i - v_o)^T \cdot n_{io} = (v_i - v_o)^T \cdot (p_i - p_o) / d_{io}$, thus, we can get that $\partial v_{io} / \partial v = n_{io}$ and $\partial v_{io} / \partial p = (v_{io} n_{io} -$

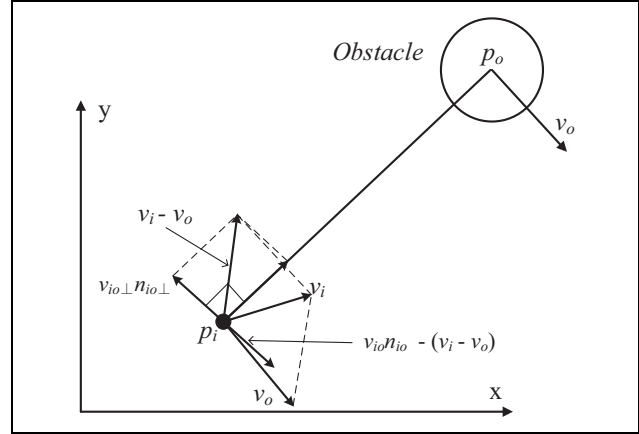


Figure 4. Vectors for construction of the repulsive potential field.

$(v_i - v_o) / d_{io}$ according to the study in Ge and Cui.²⁰ Since $v_{io} n_{io} = (v_i - v_o) \cdot n_{io}$, thus $\partial v_{io} / \partial p = -v_{io} n_{io} / d_{io}$, where n_{io} is the unit direction vector from USV to the obstacle and $n_{io\perp}$ is the unit vector perpendicular to n_{io} .

The time to perform the OA action is also important for both the refinement and the safety of the path.²⁸ We set the safe distance as $r_o = v_{io}^2 / 2 a_m$, where a_m is the maximum deceleration, thus USV has enough space for OA. We set the repulsive force to be infinity if $d_{io} \leq r_o$ for emergency OA. The repulsive force is calculated by formulas (6) and (7), and $d_{io} = ((x_i - x_o)^2 + (y_i - y_o)^2)^{1/2}$

$$\frac{\partial \phi_{OA}}{\partial v_{io}} \cdot \frac{\partial v_{io}}{\partial v} = \begin{cases} \frac{d_{it}^2 v_{io}}{a_m r_o^2} \cdot \left(\frac{1}{d_{io}} - \frac{1}{r_o} \right) \cdot n_{io}, & \text{if } d_{io} \leq R_o \text{ and collision may happen} \\ 0 & , \text{ otherwise} \end{cases}$$

$$\frac{\partial \phi_{OA}}{\partial d_{io}} \cdot \frac{\partial d_{io}}{\partial p} = \begin{cases} \frac{d_{it}^2}{r_o^2} \cdot \left(\frac{1}{d_{io}} - \frac{1}{r_o} \right) \cdot n_{io}, & \text{if } d_{io} \leq R_o \text{ and collision may happen} \\ 0 & , \text{ otherwise} \end{cases}$$

$$\frac{\partial \phi_{OA}}{\partial d_{it}} \cdot \frac{\partial d_{it}}{\partial p} = \begin{cases} d_{it} \cdot \left(\frac{1}{d_{io}} - \frac{1}{r_o} \right)^2 \cdot n_{it}, & \text{if } d_{io} \leq R_o \text{ and collision may happen} \\ 0 & , \text{ otherwise} \end{cases}$$

$$\frac{\partial \phi_{OA}}{\partial v_{io}} \cdot \frac{\partial v_{io}}{\partial p} = \begin{cases} -\frac{d_{it}^2 v_{io}}{a_m d_{io} r_o^2} \cdot \left(\frac{1}{d_{io}} - \frac{1}{r_o} \right) \cdot v_{io} n_{io\perp}, & \text{if } d_{io} \leq R_o \text{ and collision may happen} \\ 0 & , \text{ otherwise} \end{cases} \quad (7)$$

Vectors for the construction of the repulsive potential field are shown in Figure 4. Thus, the repulsive force

component along n_{io} pushes the USV away from obstacles, and the component along $n_{io\perp}$ provides steering torque for

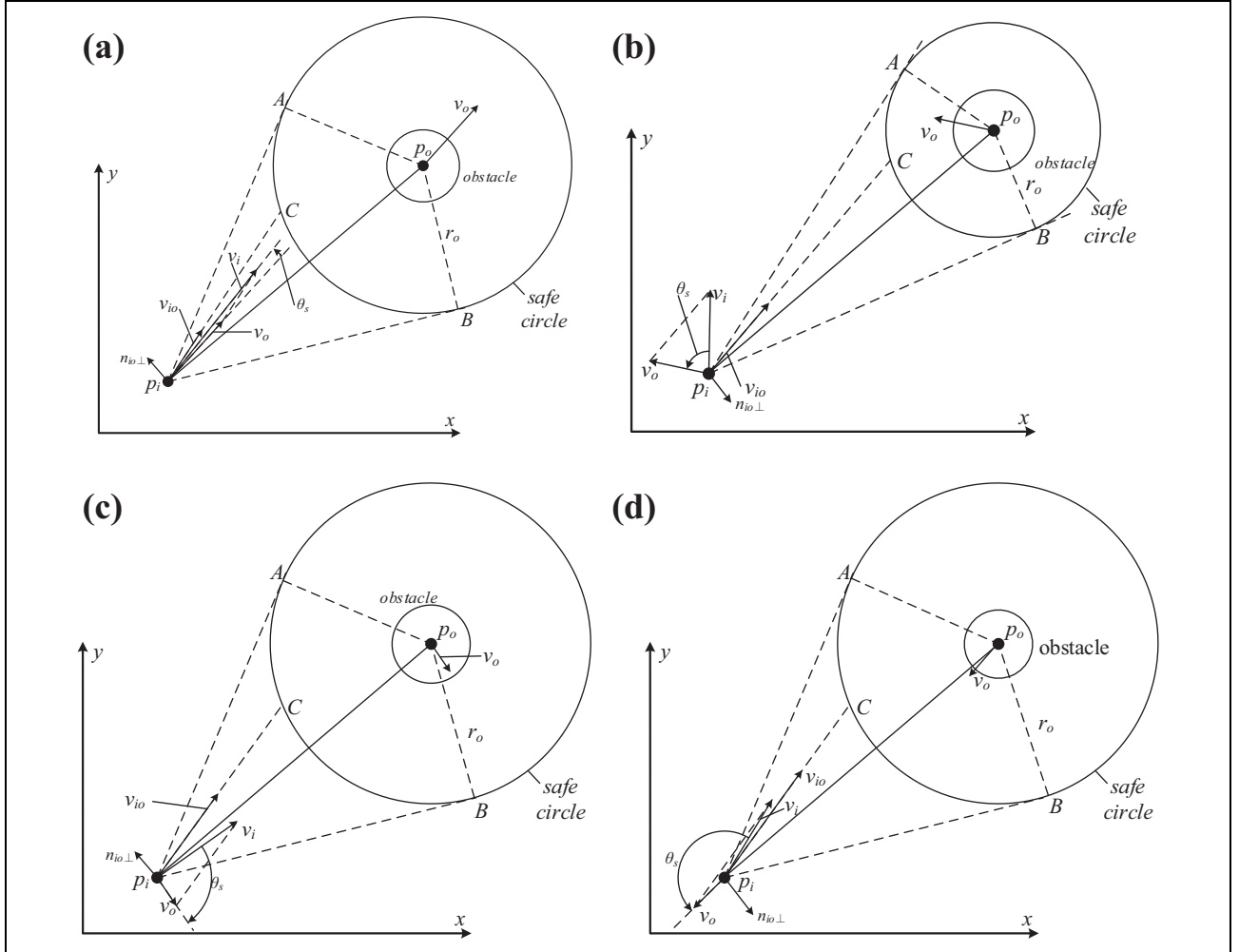


Figure 5. Typical dynamic OA situations according to the COLREGs. (a) USV overtakes DO. (b) DO crosses the course of USV from the right. (c) DO crosses the course of USV from the left. (d) USV and DO move head-on. OA: obstacle avoidance; COLREGs: the International Regulations for Preventing Collisions at Sea; USV: unmanned surface vehicle; DO: dynamic obstacle.

USV to avoid obstacles. We add a positive or negative weight on $n_{io\perp}$ to change the direction of the steering vector according to the COLREGs.

If the relationship between DO and USVs is cooperative, then USV chooses the OA direction according to the COLREGs and the intention of DO action that is assumed to be given. Such a relationship exists between USV members in a formation or the communicated USVs and DOs. Then, we add the positive or negative sign to $n_{io\perp}$ to determine the direction of the repulsive force in terms of the absolute angle (θ_s , $0 < |\theta_s| < \pi$) that is from the motion direction of USV to that of DO. If $|\theta_s| = 0$ or π , then we adjusted v_o by a small angle clockwise. Thus, according to the COLREGs, the direction of $n_{io\perp}$ is the steering direction of USV that enlarges $|\theta_s|$. Figure 5 shows four typical OA situations of USV. Then, the OA direction and the repulsive force are determined. If the relationship between USVs and DO is not cooperative, then no rules should be complied with, and USV avoids DO by a low-cost path.

OPP algorithm for USVs

The potential field of the global environment is constructed according to the given environmental information beforehand. This procedure is regarded as environment preprocessing for online burden reduction. The local potential field should be rebuilt during online planning if the environmental information changes or a DO occurs. The online built local potential field may have the local optimal problem. Thus, the randomness of the RRT* extension is kept to guarantee probabilistic completeness. This feature of RRT* can help the planner escape from the possible local optimum of the local potential field.

Local potential field modeling

The traditional APF method handles the obstacles with irregular shapes by two schemes, i.e. the circular bounding box method and the decomposition method. The bounding box method considers the obstacles by the circular bounding

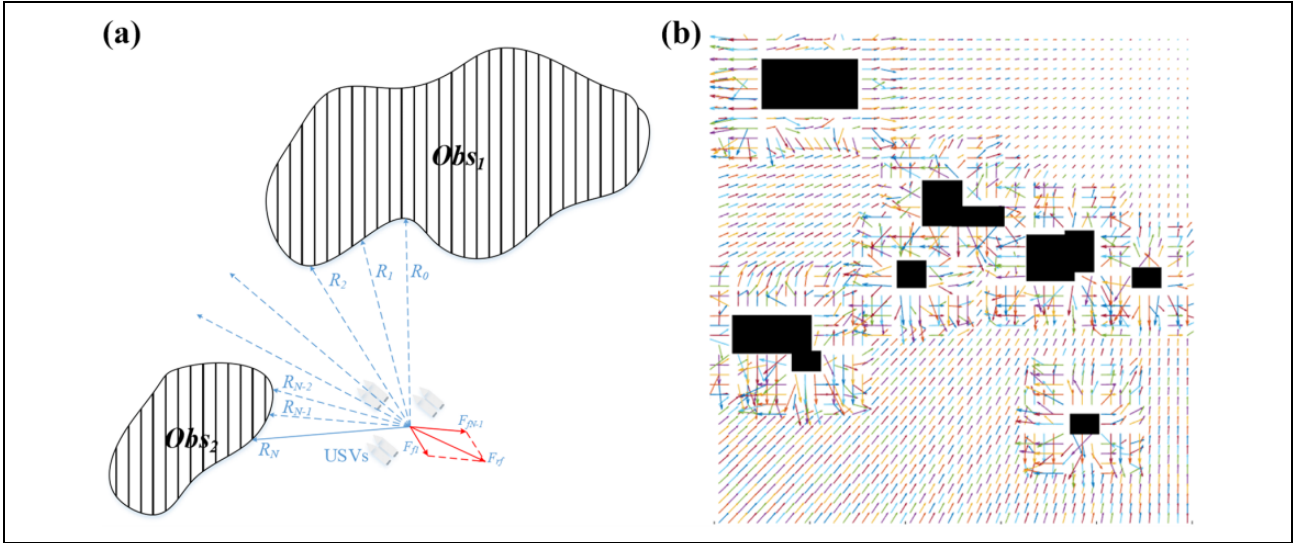


Figure 6. Local environmental potential field construction. (a) Repulsive force from irregular-shaped obstacles. (b) Expressive potential field vectors.

boxes, and then, the distance between USV and obstacle is easy to compute. However, if the obstacle is thin and long, the circular bounding box method is rather inaccurate. The decomposition method splits the irregular shape obstacle into unit circulars, but the decomposition process is complicated, and the method intends to only reflect the local features of obstacles.

Front-facing scanning radar is a sort of crucial sensor for OA, thus if the local potential field is constructed according to the characteristics of radar data, then the virtual forces can be calculated directly by the radar readings efficiently. The distances and obstacle's occupied area information that is crucial for OPP are easy to be acquired from radar readings.²³ Meanwhile, OPP generally needs the information on obstacles in front of USVs rather than that behind USVs. Thus, the radial modeling method is applicable. The local environment is expressed in vector form as follows: $R = [R_1, R_2, \dots, R_N]$ according to the study in Ge et al.²³ As shown in Figure 6(a), $R_i = \min\{R_r, R_d\}$, ($i = 1, 2, \dots, N$), where R_d is the minimum distance between USV and obstacle on a radar beam and R_r is the radius of the sensing range.

Since consecutive radar readings can be categorized as the feedback of the same obstacle. As shown in Figure 6(a), R_0, R_1 , and R_2 are considered as the readings of *Obs1*, and R_{N-2}, R_{N-1} , and R_N are considered as the readings of *Obs2*. Then, the relatively closest reading from the same obstacle can be used to calculate the repulsive potential field, for instance, R_1 is used to compute the repulsive potential field from *Obs1* because R_1 returns the minimum distance among all the readings from *Obs1*. For the same reason, R_{N-1} can be used to calculate the repulsive potential field from *Obs2*. Then, the repulsive force F_{rj} from obstacles is calculated as the resultant force of F_{r1} and F_{rN-1} .

The direction of the k th reading is denoted by $n = [\cos\theta(k) \sin\theta(k)]^T$, where $\theta(k) = 180 - (k-1)\theta_s$, and the motion

direction of USV is defined as the reference direction of zero degree angle, θ_s is the angle between a pair of adjacent readings, and θ_s is determined by the angular resolution of the sensor. Figure 6(b) is an expressive potential field vector constructed by the proposed method.

Cooperative potential function-based RRT* method

Leveraging the advantages of the baseline APF and RRT*, a cooperative potential function-based RRT* (CPRRT*) scheme is proposed. The new cooperative potential function is used to guide the exploration and exploitation processes of RRT*, eventually improving efficiency. The RRT* method is used to handle the motion constraints and the possibly unstructured and changing information to plan paths online. The potential functions are used to directionalize samples to promising areas where the optimal path more probabilistically exists, to reduce the number of iterations. The samples adjustment and path tree rewiring of CPRRT* are regarded as the exploitation or refinement process for enhancing the energy efficiency of paths.

The potential function-based sampling process and totally random sampling process are performed alternately by a probabilistic threshold t_p to keep the balance between exploitation and exploration. The description of the OPP algorithm is listed below.

The following steps are conducted iteratively until the OPP time expires.

1. A sample q_s is spawned randomly according to the uniform probability distribution in the sensing domain of USV, and the nearest node (q_n) to the sample q_s is searched on the path tree.
2. A random number $rand \in [0, 1]$ is generated. If $rand$ is lower than the probability t_p , then the potential

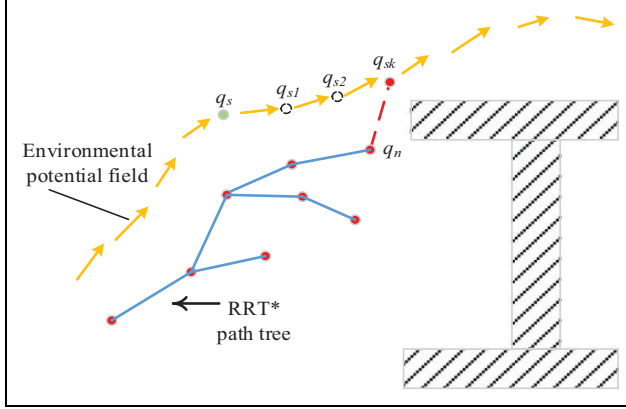


Figure 7. Illustration of the sample adjustment.

function-based sampling scheme is performed, else, the random sampling procedure is conducted. The potential function-based sampling scheme is as follows. The sample q_s is directionalized by following the cooperative potential field for k (set as 3) times if the adjustment is collision-free. If a collision happens, then the directionalizing process will stop and the last collision-free sample during the adjustment will be kept.

3. The path tree node q_n is extended toward q_s . If the extension succeeds, then a new node is added to the path tree and the locally topologic path tree rewiring process is conducted. The extension step length is set as $\eta = v_U \Delta t$, where v_U is the estimated velocity of USV at q_n and Δt is the control step duration. The expected motion direction of USV coincides with the extension direction.

Figure 7 illustrates the sampling adjustment where q_s is the original sample, and q_{s1} , q_{s2} , and q_{sk} ($k=3$) are the directionalized samples.

ALG.1 Online path planning method for USVs

Input: Global reference path; Path tree $T = \{V, E\} \leftarrow \emptyset$

```

01: WHILE Planning does not expire DO
02:   Select the subgoal for local path planning;
03:   WHILE Online planning does not expire DO
04:     Environmental information update;
05:     Radar-based local environment modeling;
06:     Random sample  $q_s$  spawning and  $rand \in^{0,1}$  generation;
07:     IF  $rand \leq t_p$  THEN
08:       CPRRTStarExtension( $q_s, T, P_{att}, P_{rep}, V_{cur}$ );
09:     ELSE /*IF  $rand > t_p$  */
10:       RRTStarExtension( $q_s, T$ );
11:   Local path querying and simplification, then smoothing;
12:   Virtual leader path following and USVs cooperative paths
   planning;

```

Algorithm implementation

ALG.1 shows our OPP algorithm. The online architecture is constructed based on the MPC framework. After a subgoal is selected, the local planning iteration starts and runs in the predefined local planning time. The USVs sail by following the currently available path, and the environmental information in the sensing range of the USVs is updated at each path extension iteration. If the local planning time runs out, then a local path is queried, and the next local planning iteration begins. The local planning with the sensing range rolls forward until the USVs achieve the goal.

At the beginning of each local planning iteration, a subgoal for the current local planning is selected, as shown by the second line, to leverage the given global reference path. If the current sensing range of USVs intersects with the global path, then the furthest waypoint to the USVs on the global path in the sensing range is selected as the subgoal. If the sensing range does not intersect with the global path, then the sensing boundary is divided into discrete points, and the discrete point nearest the global path is selected as the subgoal.

In the *fifth* line, the repulsive potential forces from obstacles are calculated according to the characteristics of the radar data. The *eighth* line shows the path tree extension of RRT* guided by the cooperative potential field. The *tenth* line shows the baseline RRT* extension process.

In the *eleventh* line, a local path is queried after the local path planning iteration expires, then the redundant waypoints are reduced, and the simplified path is smoothed via the Dubins curve. That is because the OPP method may plan a large number of waypoints, thus, the planned path is probably complex and zigzagged. To make the path easier to follow, the path should be simplified to reduce redundant waypoints and turns. We first classify the peaks of turns on the paths. If the turning angle is big, then we attempt to delete the peak by connecting the points before and after the peak. If the point is the peak of consecutive turns, then we also attempt to delete the point. The redundant waypoints reducing process is conducted sequentially until the

ALG.2 RRTStarExtension(q_{rs}, T)

Input: Sample, q_{rs} ; Cost weights matrix, W ;

Cooperative potential field, P_E ; Current vectors, V_{cur} ;
Path tree, $T \leftarrow \{V, E\}$; Path tree nodes, V ; Path tree
branches, E

```

01:  $S_{near} \leftarrow$  NearbyNodes( $T, q_{rs}$ );
02:  $S_{near.sort}()$ ;
03:  $q_n =$  NearestNode( $T, q_{rs}$ )  $\neq$  Null
04: IF CollisionFree( $E_{new} =$  Extend( $q_n, q_{rs}$ )) THEN
05:    $T \leftarrow$  Insert( $q_{new}, E_{new}$ );
06:   FOR  $q_j = S_{near.pop}()$  /*locally path tree rewiring loop */
07:     IF  $cost(q_{new}) + C(q_{new}, q_j) < cost(q_j)$ 
08:       &&  $q_{new}$  can extend to  $q_j$  THEN
09:          $E.remove(q_j.parent(), q_j)$ ;  $E.add(q_{new}, q_j)$ ;
09:   RETURN(True);
10: RETURN(False); /*no extension succeeds*/

```

end of the path. In the *twelfth* line, USVs cooperative paths are planned by following the virtual leader of USVs.

ALG.2 shows the extension process of the baseline RRT*. In the *first* line, the RRT* method finds the near nodes of q_{rs} within a ball of the radius: $R_n := \gamma \cdot (\log n/n)^{1/d}$, where $\gamma > (2(1+1/d))^{1/d} \cdot (\mu(X_{free})/\zeta_\beta)^{1/d}$ by means of the requirement of asymptotical optimality of RRT*, d is the dimension of the planning space (3), n is the number of path tree nodes, and $\mu(X_{free})$ and ζ_β denote the volumes of the obstacle-free space and the unit ball, respectively. In the *second* line, the path tree nodes in the stack of S_{near} are sorted in ascending order by their distances to q_{rs} . In the *third* line, the node q_n is the nearest path tree node of q_{rs} . E_{new} in the *fourth* line and q_{new} in the *fifth* line are the newly added path tree edge and tree node after the path tree extension, respectively.

The path tree locally rewiring (refinement) iteration is shown from the *sixth* line to the *eighth* line. In the *sixth* line, the function $\text{pop}(\cdot)$ returns the top element of the stack S_{near} . In the *seventh* line, the *cost* value of a node q_j means the sum of all the costs of path sections from the tree root to q_j , and the *C* value is the cost of a path section. The *cost* and *C* values are computed by means of formula (8). If the cost of a node q_j decreases when q_j takes q_{new} as the father node and q_j can extend to q_{new} by a collision-free path, then q_j is rewired to q_{new} and takes q_{new} as its new father, and the path section from the original father of q_j to q_j is deleted.

The cost function is defined by consideration of the path execution difficulty, energy expenditure, and safety, in terms of the path length, and the compliance of the path to the potential field or current vector

$$C(q_i, q_j) = w_l \cdot \sin(\alpha_T/2) + w_y \cdot C_y + w_d \cdot \text{dist}(q_i, q_j)$$

$$\text{s.t. } C_y = \cos(\pi - \alpha_y), \quad 0 \leq \alpha_y \leq \pi/4 \text{ or } 3\pi/4 \leq \alpha_y \leq \pi$$

$$C_y = \sin \alpha_y, \quad \pi/4 \leq \alpha_y \leq 3\pi/4$$
(8)

where α_T is the angle between the vector from q_i to q_j (marked as τ_{ij}) and the previous path section vector toward q_i , and $0 \leq \alpha_T \leq \pi$, the consideration of α_T is to reduce the numbers and aggressiveness of turns. The angle between τ_{ij} and the direction of potential field or current vector is denoted as α_y , $0 \leq \alpha_y \leq \pi$. The task-specific weighting coefficients are tuned to be $w_l = w_y = 1$ and $w_d = 0.2$, related to the energy costs by the actions of moving straight, turning, and conquering the impeding of current. Since the formula is used to compute local cost, and the distance between q_i and q_j is generally one extension step length that is generally less than 5 m in the study, thus the weights can guarantee the action cost items to be comparable. The function of $\text{dist}(\cdot)$ is used to compute the distance between two nodes.

The current can significantly influence the USV motion, thus it should be considered during the OPP procedure. Since the horizontal current flow can be regarded as

stationary and uniform, locally, thus we can consider it by the angles (denoted by $\alpha_y \in [0, \pi]$) between the direction of USV motions and those of the current, to improve the energy efficiency of the path.⁵⁻⁸

When $\alpha_y < \pi/4$, $\cos(\pi - \alpha_y)$ is negative, and the downstream path helps to save fuel for the sailing. When $3\pi/4 \leq \alpha_y \leq \pi$, the countercurrent path segment may be fuel-costly. Since the ability to resist transverse current interference of USV is weak, thus we consider the interference of the transverse current by $\sin \alpha_y$, when $\pi/4 \leq \alpha_y \leq 3\pi/4$. The OPP process is guided by α_y to plan for energy-efficient paths, such that most sections of the path are down-

ALG.3 CPRRTStarExtension($q_s, T, P_{att}, P_{OA}, V_{cur}$)

Input: Randomly spawned sample, q_s ; Path tree, T ;
Attractive potential field, P_{att} ;
Repulsive potential field, P_{rep} ;

```

01:  $q_{rs, 1} = q_s$ ;
02:  $i = 1$ ;
03:  $d_{min} = \text{NearestObs}(q_{rs, i}, X_{obs})$ ;
04: WHILE ( $d_{min} > r_o$  &&  $i \leq k$ ) DO
05:   IF  $d_{min} \leq R_o$  THEN
06:      $P_{total} = \text{PVecSyn}(P_{att}, P_{rep}, q_s)$ ;
07:   ELSE IF  $d_{min} > R_o$  THEN
08:      $P_{total} = \text{PVecSyn}(P_{att}, 0, q_s)$ ;
09:    $F_{total} = \text{Fcal}(P_{total})$ ;
10:    $i = i + 1$ ;
11:    $q_{rs, i} = q_{rs, i-1} + \lambda \cdot (F_{total} / \|F_{total}\|)$ ;
12:    $d_{min} = \text{NearestObs}(q_{rs, i}, X_{obs})$ ;
13: IF ( $d_{min} \leq r_o$ ) THEN
14:   RRTStarExtension( $q_{rs, i-1}, T$ );
15: ELSE /* $k$ */
16:   RRTStarExtension( $q_{rs, i}, T$ );

```

stream, and just a small amount of sections are transverse-current ones.

ALG.3 shows the potential function guided path tree extension process. In the *first* line, q_s is assigned to $q_{rs, 1}$ as the initial sample. In the *third* line, d_{min} denotes the minimum distance between the sample $q_{rs, i}$ to obstacles. In the *fourth* line, r_o is the minimum safe distance between USV and obstacles, and k (3) is the fixed times of adjustment. In the *fifth* line, R_o is the reactive radius for OA. In the *sixth* and *eighth* lines, the function $\text{PVecSyn}(\cdot, q_s)$ is defined for synthesizing the potential vectors at q_s . If the sample is outside the OA reactive range, then the synthesizing process will not consider the repulsive force from obstacles. In the *ninth* line, $\text{Fcal}(\cdot)$ calculates the resultant force by means of P_{total} . In the *eleventh* line, the sample is directionalized for a distance of λ along the direction of F_{total} , where λ is the extension step length of RRT*. In the *thirteenth* line, the condition means the adjusted sample is within the minimum distance from obstacles, then we extend the path tree toward the last adjusted collision-free sample $q_{rs, i-1}$. The *fifteenth* and *sixteenth* lines mean the adjustment process is conducted

for k times, and we extend the path tree toward the adjusted sample $q_{rs, i}$.

Algorithm analysis

The heuristic sampling process is used to enhance the efficiency of RRT* because the computational complexity of generating samples is far lower than that of the path extending, and high-quality samples can accelerate the converge process of RRT* to optimal paths.⁴⁵

The qualitative illustration of the probabilistic completeness of CPRRT* is listed below. The path tree of CPRRT* is a connected tree rooted at the initial state of USV, and the path tree extends toward samples from their nearest tree nodes. The potential field is just used to accelerate the path searching process of RRT* by guiding samples to avoid obstacles and toward a goal probabilistically.^{43–46} Moreover, the samples remain probabilistic randomness to explore the whole space.

CPRRT* converges to high-quality paths, and the formal proof is as follows. A collision-free path τ^* is said to be optimal in terms of Euclidean distance, if it has weak δ -clearance, meaning that the path can be deformed by a collision-free homotopy $h(y)$ to another path τ with bigger distances than δ from obstacles; τ has the same start and end points to τ^* ; and τ^* has distance no bigger than δ from obstacles.³⁵ The path τ^* is provided with the set of all feasible paths such that $C(\tau^*) = \operatorname{argmin}_{\tau \in \text{feasible}} \{C(\tau)\}$, where feasibility means that paths satisfy motion constraints of USV and are collision-free as well as not violating any OA rule. CPRRT* performs exploration and exploitation of configuration spaces to refine paths. The heuristics of CPRRT* guide paths toward the weak δ -clearance by directing samples to spaces with a high probability of containing an optimal path. The potential field continues directing samples down the slope of the potential field to the obstacle-free goal region until the sample gets very close to obstacles or the threshold is reached. The algorithm parameters are set to be the same as RRT* except that the rewiring process also considers the current to adjust the path toward the energy-saving direction. Therefore, the OPP solution of CPRRT* can converge to a high-quality path that considers the path length, ease of following, and the influence of current on the USV motion, comprehensively.

However, path optimality is hardly achieved online due to the limited time, and efficiency is crucial for an online planner. To balance the complexity against efficiency, a heuristic potential field function guided sampling scheme is devised for leveraging the known environmental information. Therefore, CPRRT* is probably more efficient than the baseline RRT* method.

Potential field for cooperative paths planning

After a path section is planned online, the virtual leader is supposed to follow the path. Thus, the position and velocity

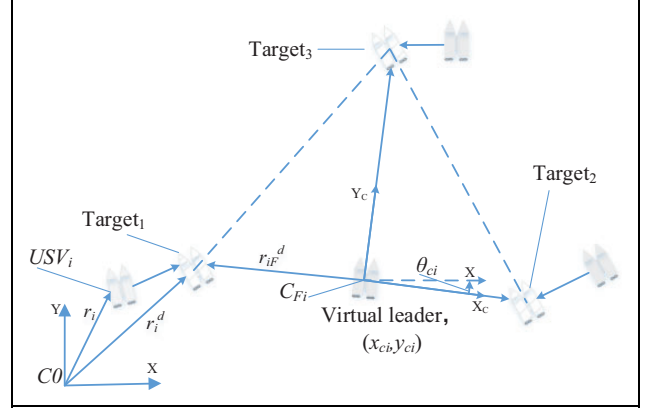


Figure 8. Illustration of the cooperative motion of USVs. USV: unmanned surface vehicle.

of the virtual leader can be calculated in real time. Then, the path planning targets of USVs are estimated.

We define the inner potentials to keep the USVs formation and then the formation keeping potential field and the OA potential field are synthesized. The USVs in the triangle formation are illustrated in Figure 8, where C_0 is an inertial coordinate system, C_{Fi} denotes the formation coordinate system, and the coordinate (x_{ci}, y_{ci}) of the virtual leader in $\{C_0\}$ is the origin of $\{C_{Fi}\}$. We define $p_i = (x_i, y_i)$ and $p_i^d = [x_i^d, y_i^d]^T$ to be the actual position vector and the motion target vector of USV_i , respectively, and $p_{iF}^d = [x_{iF}^d, y_{iF}^d]^T$ is the position vector of USV_i with respect to the virtual leader. We can adjust p_{iF}^d for various formations. The transformation of p_{iF}^d to p_i^d is defined as follows

$$\begin{bmatrix} x_i^d(t) \\ y_i^d(t) \end{bmatrix} = \begin{bmatrix} x_{ci}(t) \\ y_{ci}(t) \end{bmatrix} + \begin{bmatrix} \cos[\theta_{ci}(t)] & -\sin[\theta_{ci}(t)] \\ \sin[\theta_{ci}(t)] & \cos[\theta_{ci}(t)] \end{bmatrix} \begin{bmatrix} x_{iF}^d(t) \\ y_{iF}^d(t) \end{bmatrix} \quad (9)$$

The ring-shaped formation keeping potential is constructed around each USV according to the studies in Ge and Cui²⁰ and Qin et al.²⁸ The attractive potential field is constructed in the following formula

$$U_{att}(p, v) = 0.5 \cdot \alpha_p \cdot (p_{tar} - p_i)^2 + 0.5 \cdot \alpha_v \cdot (v_x - v_i)^2 \quad (10)$$

where p_{tar} denotes the position of the target, v_x denotes the velocity of the virtual leader, p_i and v_i are the actual position and velocity of the USV member, and α_p and α_v are specified to be 0.7 and 0.3, respectively.

As shown in formula (11), the attractive force has two components where F_{att1} is used to follow the position of the target and F_{att2} is used to follow the velocity of the virtual leader. The direction of the unit vector n_{it} is from the position of USV to that of the motion target, and the direction of the unit vector n_{vit} is from the direction of the USV velocity to that of the virtual leader

$$\begin{cases} F_{\text{att}} = -\nabla_p U_{\text{att}}(p, v) - \nabla_v U_{\text{att}}(p, v) = F_{\text{att}1} + F_{\text{att}2} \\ F_{\text{att}1} = \alpha_p \cdot |p_{\text{goal}} - p(t)| \cdot n_{it} \\ F_{\text{att}2} = \alpha_v \cdot |v_x - v(t)| \cdot n_{vit} \end{cases} \quad (11)$$

USVs will be pushed back by the repulsive forces from other members for OA if they enter the domain of other members, to keep the activities of each USV within their own allowable range. The repulsive potential field from other USV members is defined in the following formula

$$\phi_{rij} = \sum_{i=1}^n \left(\left(\frac{1}{d_{ij}} - \frac{1}{r_o} \right)^2 \cdot d_{it}^2 \right) \quad (12)$$

where the parameters of the repulsive potential field for formation keeping are the same as those in formula (7),

and n is the number of the USV members. A USV member considers other members as obstacles, and d_{ij} is the distance between members. The steering direction of USV_i avoiding

Table 2. Partial parameters of the online path planner.

δ_c (s)	PR (km)	Δt (s)	r_o (m)	γ_{RRT^*}
5	5	0.1	5	20

PR: radius of the sensing domain; RRT*: rapidly exploring random tree.

USV_j is determined by $n_{ij\perp}$, and the calculation of the direction of $n_{ij\perp}$ is similar to that of $n_{io\perp}$ in Figure 5 according to the COLREGs. Formula (13) shows the repulsive force from USV_j that is consisted of three components, $F_{rij1}n_{ij}$, $F_{rij2}n_{ij\perp}$, and $F_{it}n_{it}$. Similar to formula (7), formula (14) is used to calculate the repulsive force component in the direction n_{ij} from USV_j to USV_i , formula (15) aims to calculate the repulsive force component in the direction perpendicular to n_{ij} , and formula (16) is utilized to compute the repulsive force component considering the attraction from the motion target

$$F_{rij} = \sum_{i=1}^n (F_{rij1}n_{ij} + F_{rij2}n_{ij\perp}) + F_{it}n_{it} \quad (13)$$

$$F_{rij1} = \begin{cases} \frac{d_{it}^2(v_{ij} + a_m)}{a_m r_o^2} \cdot \left(\frac{1}{d_{ij}} - \frac{1}{r_o} \right), & \text{if } d_{ij} \leq R_o \text{ and collision may happen} \\ 0 & \text{, otherwise} \end{cases} \quad (14)$$

$$F_{rij2} = \begin{cases} -\frac{d_{it}^2 v_{ij}}{a_m d_{ij} r_o^2} \cdot \left(\frac{1}{d_{ij}} - \frac{1}{r_o} \right) \cdot v_{ij\perp}, & \text{if } d_{ij} \leq R_o \text{ and collision may happen} \\ 0 & \text{, otherwise} \end{cases} \quad (15)$$

$$F_{it} = \begin{cases} d_{it} \cdot \left(\frac{1}{d_{ij}} - \frac{1}{r_o} \right)^2, & \text{if } d_{ij} \leq R_o \text{ and collision may happen} \\ 0 & \text{, otherwise} \end{cases} \quad (16)$$

The OA priorities of USVs are defined before sailing off to avoid the deadlock problem in the cooperative collision avoidance process between USVs. The low-priority members give way to the high-priority USVs. Once the high-

priority USV has reached the estimated position, the collision situation becomes clear. Then, the low-priority member plans for the OA path.

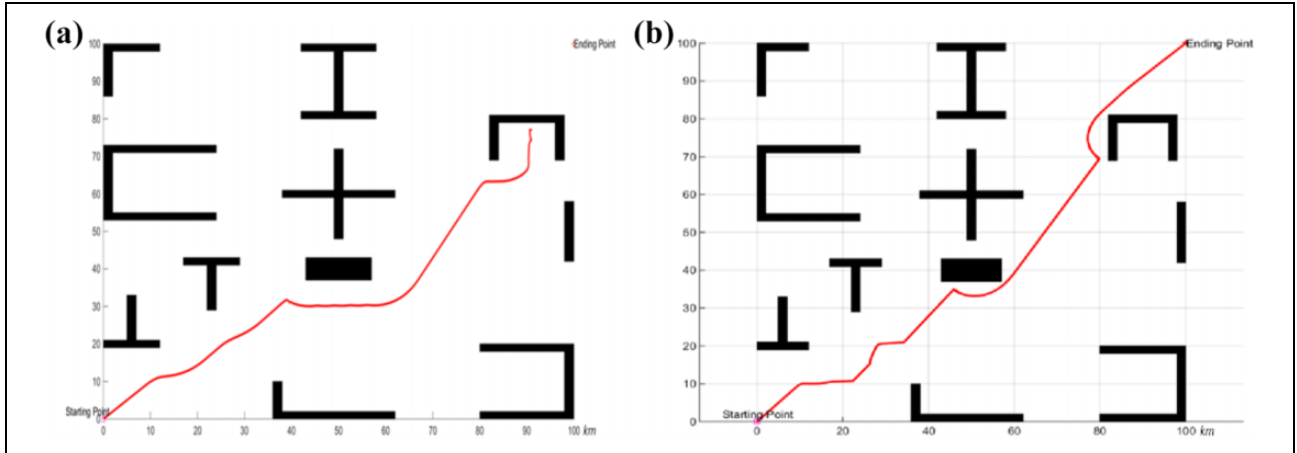


Figure 9. Path planning results of the APF method under the Bug Trap environment. (a) Planning being trapped. (b). Planning result in a Bug Trap environment. APF: artificial potential field.

The CPRRT* method with the goal bias strategy was used as the local planner. Since we considered the problems of OA and path refinement during the virtual leader path planning process, thus we combined the goal bias scheme that chooses the current target as samples within a probability threshold (0.3).⁴¹

Experimental results and analyses

Experimental environment setup

Simulations were conducted on a computer with Windows 10 64 bit OS, 16 G RAM, and Intel(R) Core(TM) i7-7820HQ CPU @ 2.90 GHz.

The OPP parameters are presented in Table 2, where δ_c is the fixed local planning time and Δt is the interval between control steps, and a local path generally contains more than 50 control steps. PR is the radius of the sensing domain, r_o is the minimum safe distance between USV and obstacles, and γ_{RRT^*} is used in a sample's near neighbors searching process of RRT*. The maximum speed of our USV is 5 m/s, the maximum translational acceleration is 1.5 m/s², the maximum lateral acceleration is 10 m/s², and the minimum turning radius is 2 m.

Ablation experiments

Expressive simulation results. We performed the APF method, baseline RRT* method, and heuristic RRT* method to understand the contribution of each component to CPRRT* performance. The units of the following figures are kilometers.

Figure 9 shows the planning result of the proposed APF method in the Monte Carlo experiment, where the locations of the obstacles are randomly changed. The speed of USV is set to be 5 m/s without obstacles nearby, and the OA speed is set to be 2.5 m/s. The direction of the velocity coincides with that of the path section.

Figure 9 indicates the possible planning results in the Bug Trap environment. The environment consists of many traps formed by irregularly shaped and concave obstacles, and it also has low visibility. The experiment aims to illustrate that the simulated environment is rather complex for APF-based and RRT*-based planners. Figure 9(a) shows that APF planning easily falls into the local optimum. Figure 9(b) indicates that aggressive turns may exist on the path. In addition, our repulsive force calculation in terms of radar readings is practical.

Figure 10 illustrates the results of the ablation experiments where the effectiveness of the heuristic modules is illustrated. The heuristics of CPRRT* include the potential field-based sample directionalizing, subgoal selection that refers to the global path, and path tree topological structure refinement by rewiring.

Figure 10(a) shows the planning results of CPRRT* with all of the heuristic modules. The red solid curve is the planned path. The green- and blue-dotted curves are the original paths captured by APF and the simplified path which is smoothed via the Dubins curve, respectively. The tree-shaped lines show the path tree branches during the path searching process. The green stars denote subgoals.

The reference path is queried along the direction of the gradient descent of the potential field. Thus, it is a high-quality path in terms of static environment information. The CPRRT* path deviates not far from the reference path. It is short with a small number of gentle turns. Most sections of the path conform to the potential vectors. Therefore, the CPRRT* path is probably a low-cost path.

The small number of tree branches illustrates that the path searching process is efficient because of the heuristics. Figure 10(a) describes the potential vectors built according to the given environment information. The vectors can guide the OPP process to collision-free spaces to improve the efficiency of the RRT*-based planner. The path tree exploitation by rewiring helps refine paths.

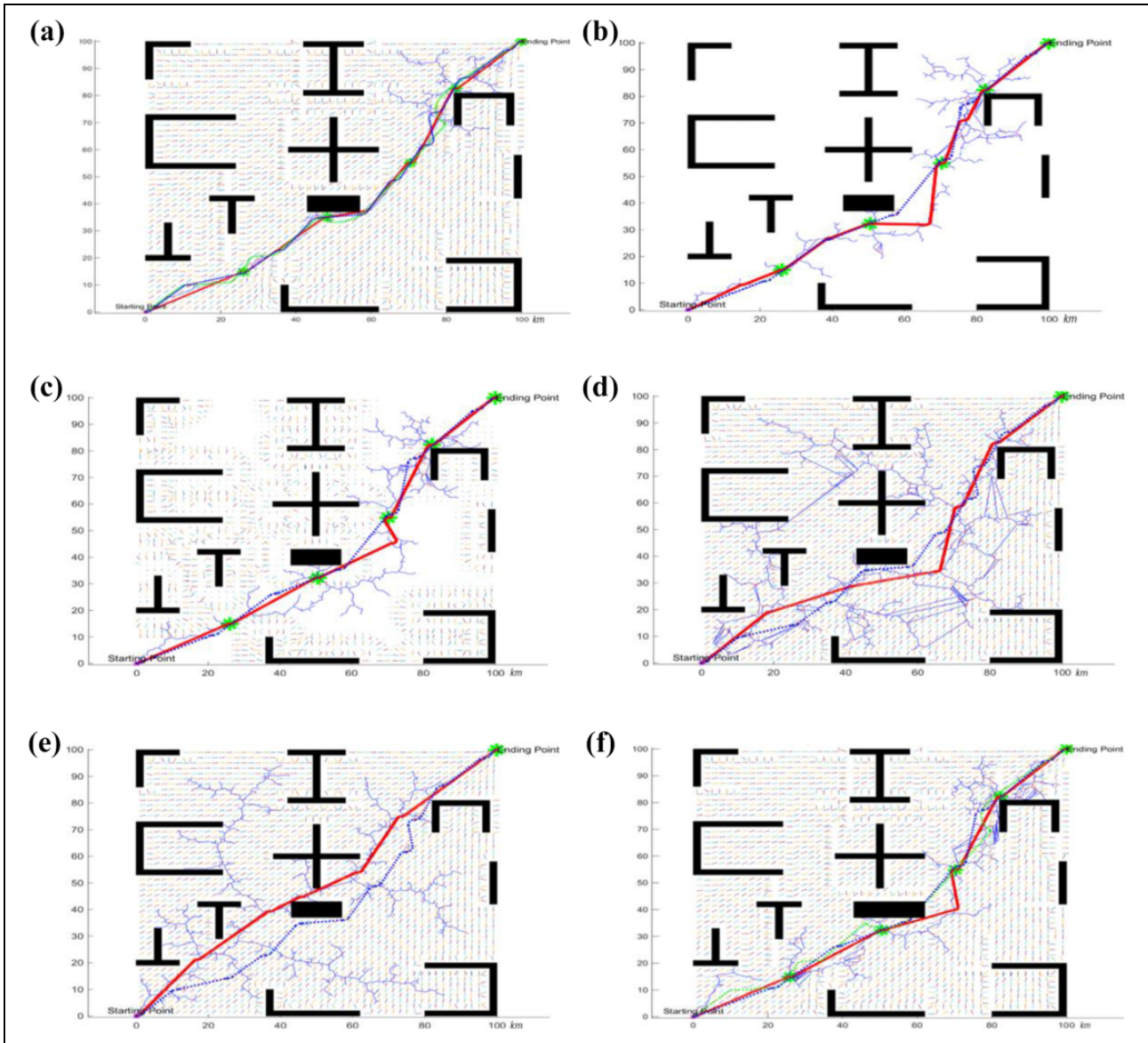


Figure 10. Expressive results with different heuristic modules. (a) Result of CPRRT*. (b) Result without potential field heuristics. (c) Planning result of the PRRT* method. (d) Planning result without reference path. (e) Result of CPRRT without rewiring. (f) Result of CPRRT* with blocked reference path. CPRRT*: cooperative potential function-based rapidly exploring random tree; PRRT*: potential function-based rapidly exploring random tree.

Figure 10(b) illustrates the result of the planner without the potential field heuristics. Compared with the path of CPRRT*, the path in Figure 10(b) deviates further from the reference path with many turns, including even the aggressive turns in the middle of the subfigure. That observation shows that the path is probably costly.

Figure 10(c) shows the planning results of the potential function-based RRT* (PRRT*) method. PRRT* uses the repulsive potential field without considering the USV velocity. Similar to the studies of literature,^{43–45} the sample adjustment does not consider the attractive force from the subgoals. The path is longer with more turns than the path in Figure 10(a). Aggressive turns also exist, as shown in

the middle of the subfigure. The probable reason is that PRRT* does not consider the attractive force from the subgoals, and the repulsive force pushes the path away from obstacles as far as possible. The path tree of PRRT* has no absolutely random sampling process. Thus, the planner may not be exploring space as comprehensively as CPRRT*. Therefore, the path quality may be lower than that of CPRRT* in dynamic environments when the planning time is quite limited.

Figure 10(d) expresses the planning result without the subgoal selection. Moreover, the planner and the attractive potential field consider only the global goal. As shown by the broadly growing tree branches, the planner attempts to

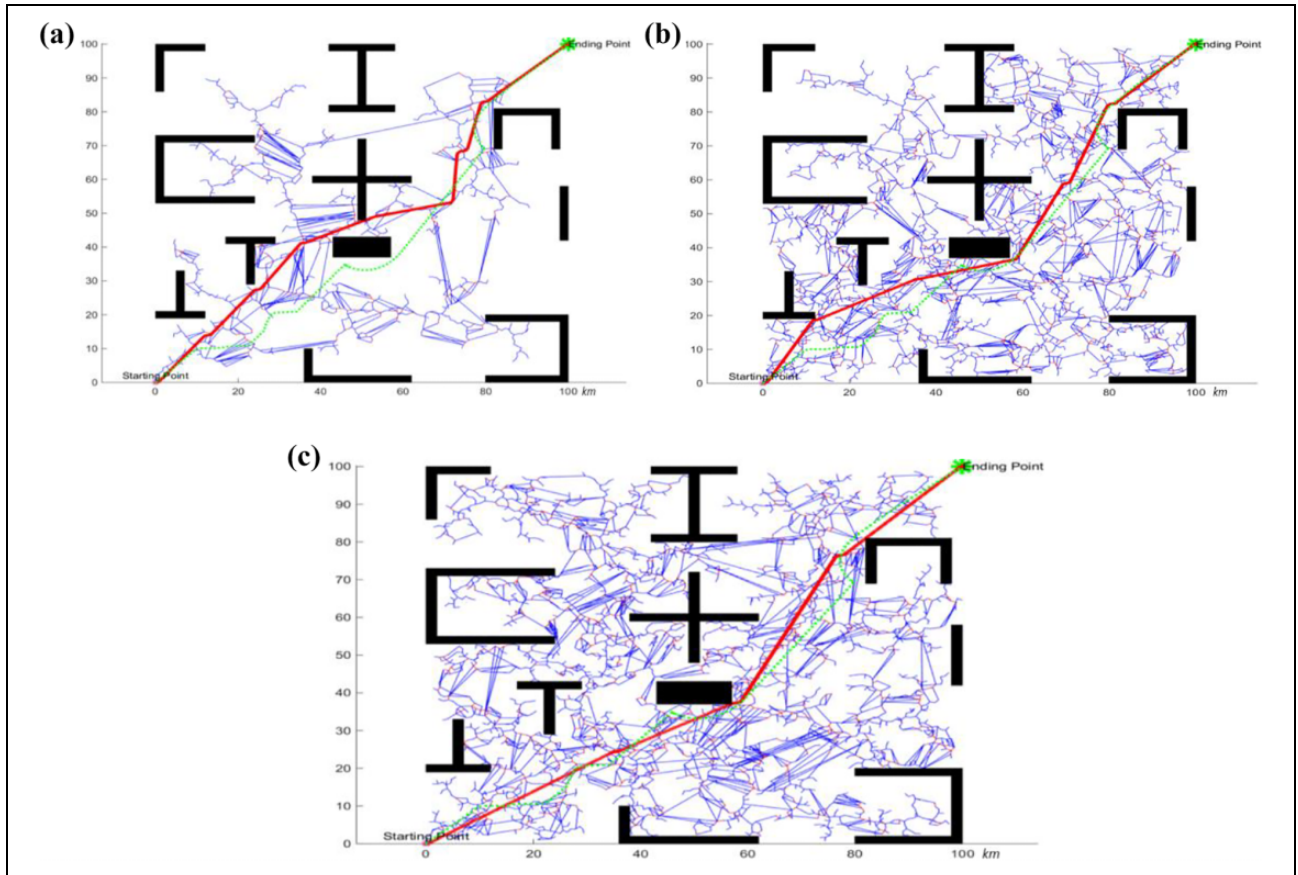


Figure 11. Expressive path refining results of CPRRT* when local iteration increasing. (a) Expressive results with 500 iterations. (b) Expressive results with 5000 iterations. (c) Expressive results with 10000 iterations. CPRRT*: cooperative potential function-based rapidly exploring random tree.

search for the path by widely exploring the space, possibly causing the path to be more costly than the path of CPRRT*. However, the planner can quickly find a path with the guidance of the potential field, as shown by the small number of branches.

Figure 10(e) describes a path of the cooperative potential function-based rapidly exploring random tree (CPRRT) method. Unlike CPRRT*, CPRRT does not refine paths by rewiring. CPRRT also does not use the guidance of subgoals.

The path of CPRRT in Figure 10(e) is probably shorter but more costly than that in Figure 10(d) because the former has more sections with larger angles with the potential vectors than the latter has. Moreover, the path has a large number of turns in the middle of the subfigure. This scenario is possible because the CPRRT has no path rewiring mechanism.

Figure 10(f) shows the result of CPRRT* when the reference path is blocked because the environmental information changed. The result demonstrates the OPP ability of CPRRT*. The online planned path can avoid the pop-up obstacle. The local potential vectors also change near the pop-up obstacle. These observations express that the proposed local potential field modeling method is practicable.

Figure 11 shows the results of CPRRT* for different local path planning iterations. The planner deleted the subgoal selection module to demonstrate the asymptotic optimality and probabilistic completeness of CPRRT*. Thus, it did not use the guidance of the reference path. The results can illustrate the relationship between planning time and resulting path quality.

Figure 11(a) shows that, although the planner can find a path within 2000 local planning iterations, the path is not a low-cost one. Figure 11(b) expresses that the resulting path is probably better than the path in Figure 11(a) because the planning iterations increase. The reason is that CPRRT* continues refining paths in the newly added time. Figure 11(c) shows that the path becomes better than the path in Figure 11(b), along with an increase in the iteration times. This observation demonstrates that the CPRRT* method has the asymptotically optimal nature derived from RRT*.

Figure 11(a) to (c) illustrates that CPRRT* can plan different path styles. As the iterations increase, the path tree explores the space extensively. Then, many kinds of paths are considered by the planner, and the resulting path is refined adoptively. The asymptotic optimality and

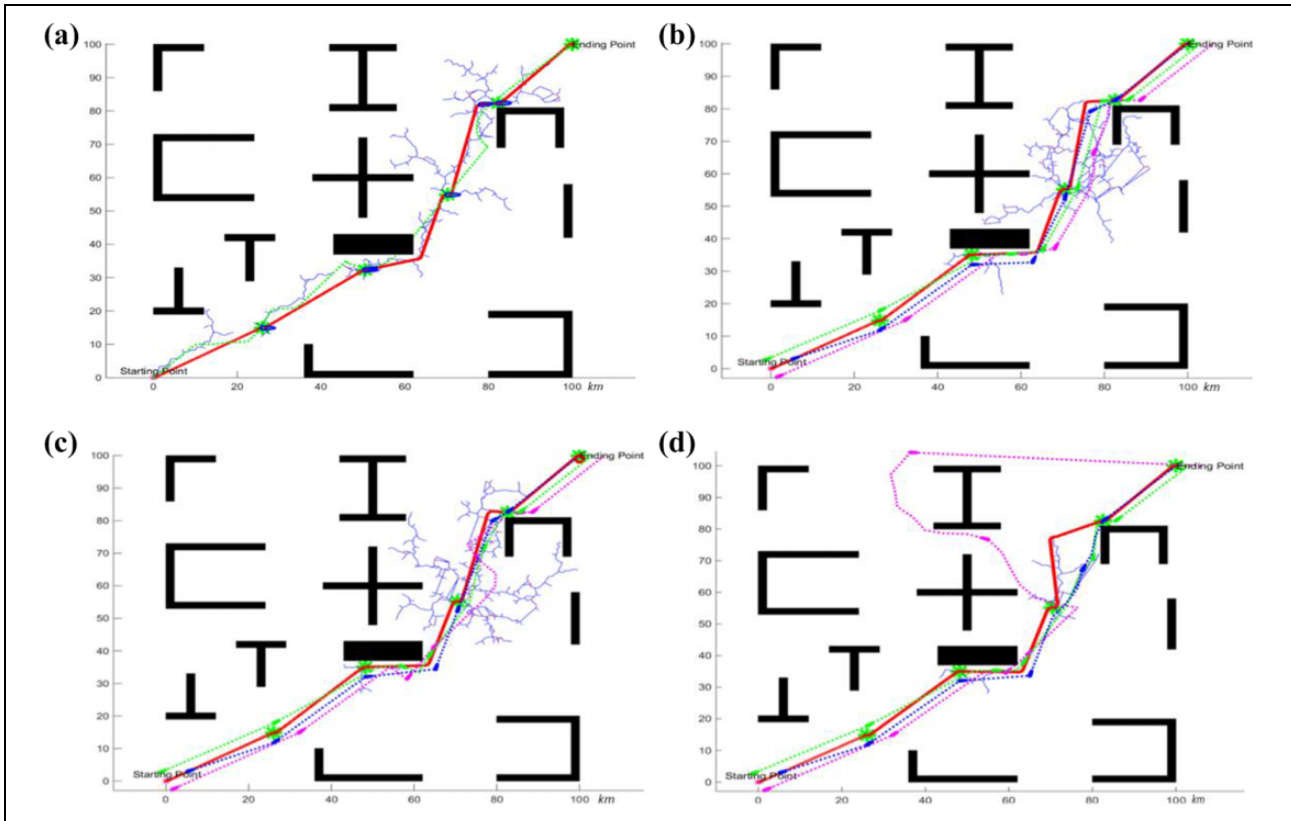


Figure 12. Illustration of cooperative paths planning process. (a) The supposed virtual leader motion. (b) USVs cooperative paths with formation transforming. (c) Planning for paths individually. (d) Planning costly path for collision avoidance. USV: unmanned surface vehicle.

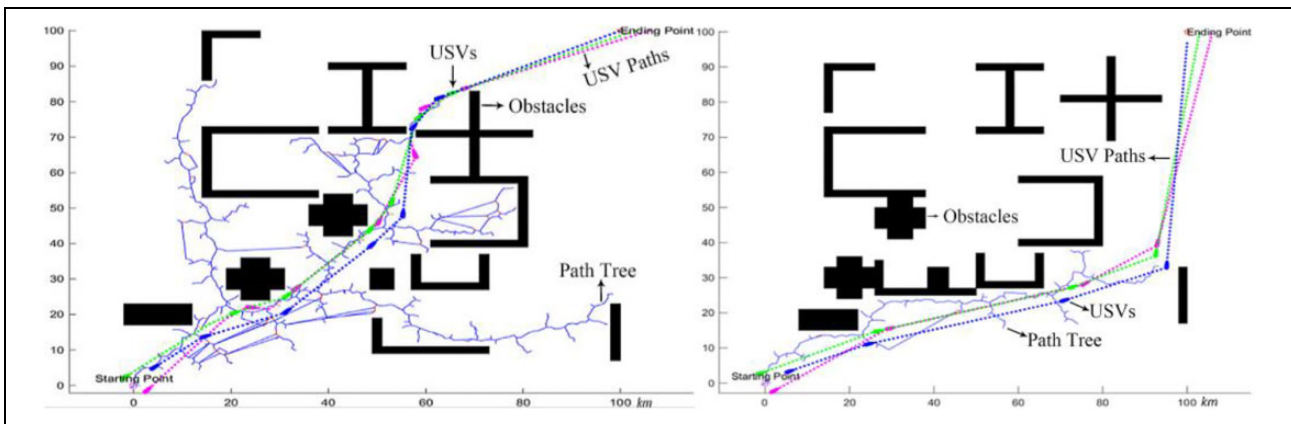


Figure 13. Monte Carlo simulation solutions in the obstacle-cluttered environment.

probabilistic completeness are kept because randomness remains in the planning process of CPRRT*.

Figure 12 illustrates the proposed cooperative paths planning process. Figure 12(a) supposes that the virtual leader moves along the online planned path, as shown by the elliptical bounding boxes. The virtual leader is supposed to be located at the boundary of the sensing domain of the USVs on the path at the beginning of each local path planning iteration. The direction of the virtual leader velocity is

estimated in terms of the geometric property of the path. We assume that the communications between USVs are available.

The planners are guided by the cooperative potential field, the defined collision avoidance priority, and the real-time USV statuses to deal with the cooperative OA problem. The dotted lines in Figure 12(b) denote the USV paths. USVs maintain the triangle formation when they are far from the obstacles. They cooperatively transform into

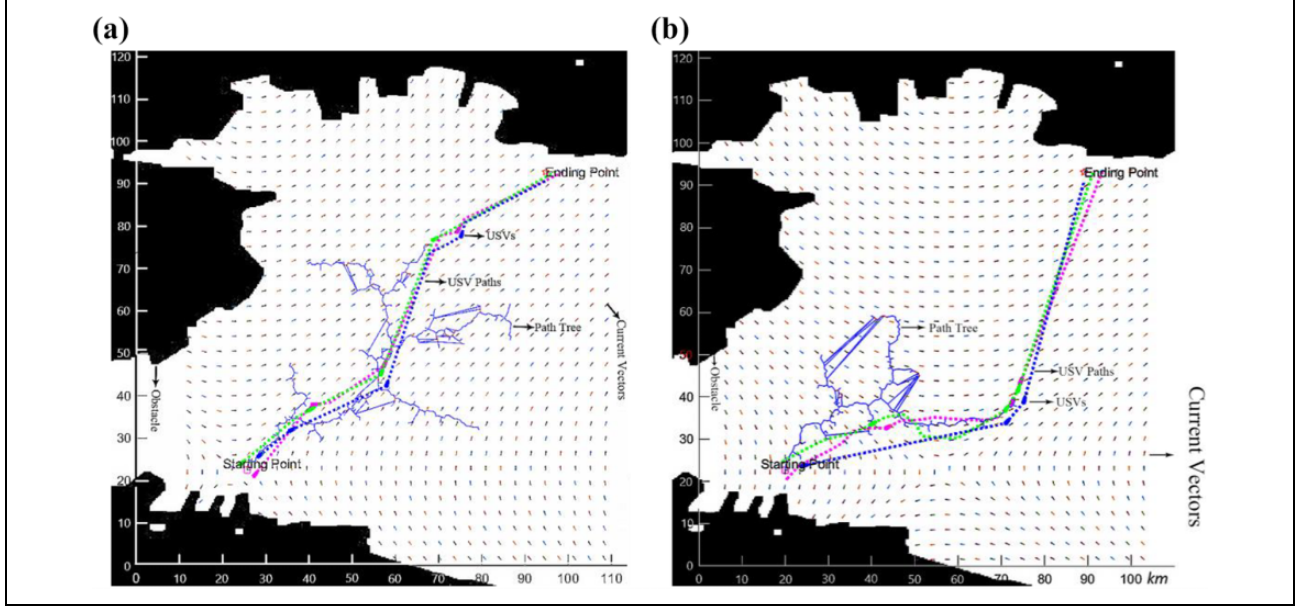


Figure 14. Path planning results with the time-varying sea current. (a) Resulting path when $t = 1$ h. (b) Resulting path when $t = 10$ h.

the linear formation to avoid collision when they become close to obstacles, as shown in the middle and the right top of the subfigure. Then, they restore the triangle formation.

The planning objective is to minimize the sum cost of all of the paths. However, the safety of the OA path is also crucial. Thus, the planners may balance safety against cost. The magenta path in Figure 12(c) describes that a long path is planned for the safety of the cooperative OA. The magenta path in Figure 12(d) also expresses that USVs avoid obstacles individually, and they can break the formation for cooperative OA. This scenario illustrates the flexibility of the distributed OPP framework.

Figure 13 shows the solutions of our planner in Monte Carlo experiments, which are employed to evaluate the efficiency and the online planning capability of the planner in dynamic environments. The cluttered, irregularly shaped obstacles with many narrow passages challenge the planner.

The online planning ability of our planner is testified for the following reasons: the planner can find OA paths according to the real-time environmental information, and the distributed planner in each USV can find a relatively low-cost path by flexibly changing the formation. The high efficiency is shown by the relatively low number of extension path-tree branches and high path quality that is certified by the few and gentle turns on paths.

A time-varying sea current is generally set to be clockwise in the northern hemisphere in Figure 14. The current simulated below refers to our study in Wen *et al.*⁴²

$$\psi(x, y, t) = 1 - \tan \left(\frac{\cos(sc_y \cdot y - B_c(t)) \cdot \cos(k_u \cdot (sc_x \cdot x - c \cdot t))}{\sqrt{1 + k_u^2 \cdot B_c^2(t) \cdot \sin^2(k_u \cdot (sc_x \cdot x - c \cdot t))}} \right) \quad (17)$$

where $B_c(t) = b + e \cdot \cos(\omega \cdot t + \theta_u)$, $b = 1.1$, $e = 0.2$, $c = 0.12$, $k_u = 0.84$, $\omega = 0.02$, and $\theta_u = \pi/2$, $sc_x = sc_y = 0.02$. The current vectors are captured by $\psi(x, y, t)$. The solution paths should consider the current vectors to save energy. Figure 14(a) and (b) illustrates the sea current model when the time (t) equals to 1 and 10 (hour), respectively.

Attempts are made to plan paths in the open sea. Thus, the primary interference of the USV navigation is the time-varying sea current. However, the real-time sea current can be measured using the current meter. The online planner should then return low-cost OA paths according to the real-time sensed sea current information. The downstream sections of a path help save energy, whereas the countercurrent or transverse-current sections add extra cost for conquering the current.

Given that the space from the planning start point and end point is set to be without obstacles, the potential field vectors are replaced using the sea current vectors in formula (8). Then, the planning is guided by the current. In practical occasions, the potential field is considered in sample adjustment if obstacles exist because the OA task is prior to the current handling task. The current is considered instead of the potential field in sample adjustment when no OA task is to be conducted. However, the CPRRT* planner always refines paths by path tree rewiring to consider the sea current.

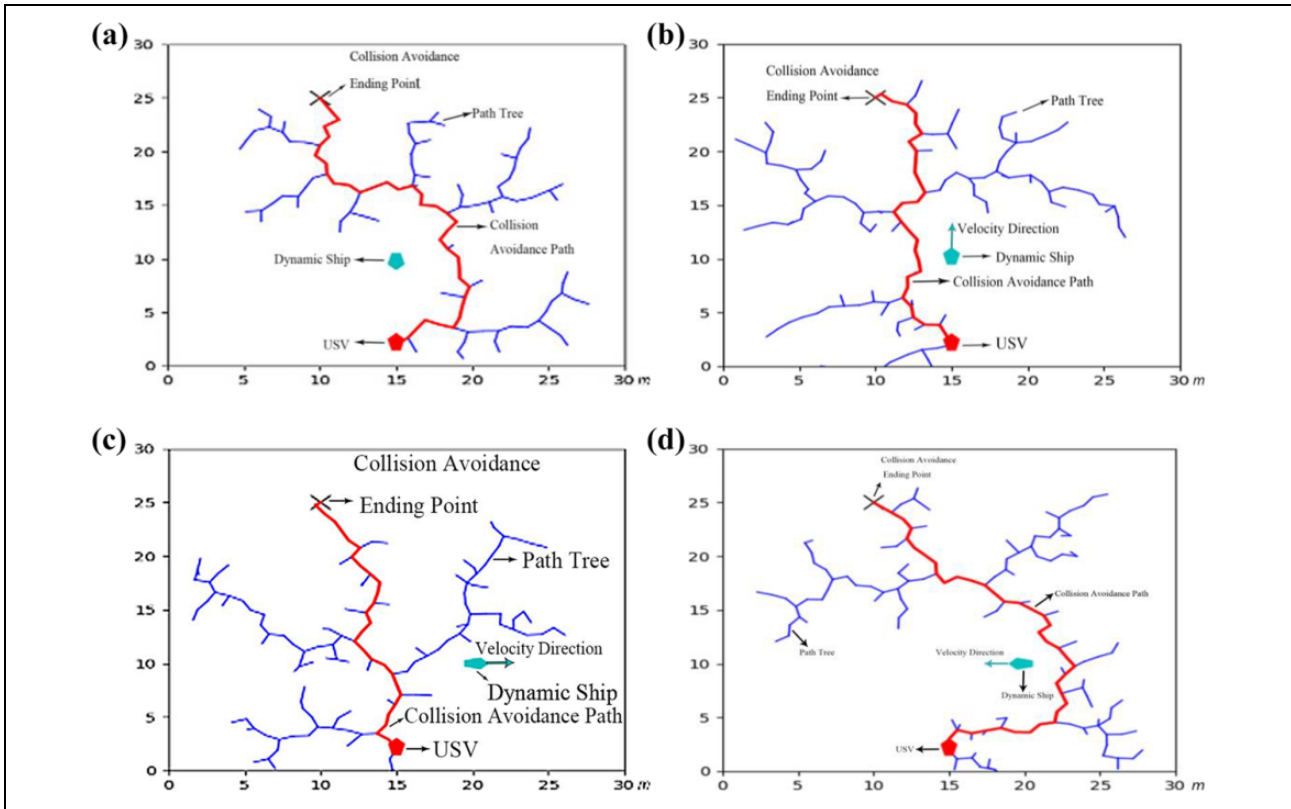


Figure 15. USV avoiding a dynamic ship in four scenarios according to COLREGs. (a) USV meeting dynamic ship. (b) USV overtaking dynamic ship. (c) Dynamic ship crossing USV from the left. (d) Dynamic ship crossing USV from the right. USV: unmanned surface vehicle; COLREGs: the International Regulations for Preventing Collisions at Sea.

Table 3. Quantitative results of the virtual leader's path planning experiments.

Algorithms	Path searching time (s)	Path cost	Path length (km)	Failure rate	Time STDEVP	Path cost STDEVP	Path length STDEVP
CPRRT*	5.1607	153.24	152.80	0.02	0.7703	3.6068	5.9433
PRRT*	5.5251	159.24	152.40	0.02	0.1763	7.7529	6.0520
CPRRT	8.0832	161.14	153.83	0.03	0.1777	14.1109	7.2461
APF	0.8181	145.05	146.67	—	—	—	—
WO potential	7.4001	158.43	151.39	0.09	1.2892	6.9518	7.7587
WO subgoal	11.5994	161.76	156.35	0.05	4.1362	4.5843	4.9417
RRT*	16.8554	174.15	170.55	0.15	16.3623	10.7744	8.5581

STDEVP: standard deviation; CPRRT*: cooperative potential function-based rapidly exploring random tree; PRRT*: potential function-based rapidly exploring random tree; APF: artificial potential field; WO: without; RRT*: rapidly exploring random tree.

In both subfigures, the paths consist of many downstream sections and few countercurrents or transverse-current sections, possibly demonstrating the online path refinement ability of our planner. That is because the quality of paths is ensured by asymptotical path refinement.

In Figure 15, the dynamic ship-avoidance problem is considered. The path extension process is guided by the potential field vectors whose directions are determined by the collision avoidance rule, namely, COLREGs. Thus, the planning is guided by the potential field to avoid the dynamic ship. Experiments are conducted in four

scenarios, as shown in Figure 15. Figure 15(a) shows the USV meets the ship. Figure 15(b) shows the USV overtakes the ship. Figure 15(c) and (d) shows the dynamic ships crossing the USV from the left and right, respectively. The polygonal red curves are the collision-avoidance paths. The red pentagons illustrate USVs, whereas the green ones illustrate ships.

The USV and ship are assumed to obey COLREGs. Then, the potential field vector directions are adjusted in terms of the encounter scenario, as indicated in Equation (7) and Figure 5, to guide the path extension directions of

Table 5. Results of planning for multiple paths.

Algorithms	Local path searching time (s)	Path cost	Path length (km)	Time STDEVP	Path cost STDEVP	Path length STDEVP
CPRRT*	0.8751	177.43	154.44	0.7223	4.8798	3.4096
PRRT*	1.4288	175.57	154.83	0.6034	9.0592	8.0825

STDEVP: standard deviation; CPRRT*: cooperative potential function-based rapidly exploring random tree; PRRT*: potential function-based rapidly exploring random tree.

Table 4. Planning results of CPRRT* with fixed local iteration times.

Iteration times	Total path planning time (s)	Path cost	Path length (km)	Time STDEVP	Cost STDEVP	Path length STDEVP
1000	1.467198067	154.9306	149.7542	0.47263971	8.504669	2.505083
5000	77.65518073	152.6351	149.4904	10.04815065	16.73168	2.120877
10,000	337.3860333	147.3090	145.7312	4.08819589	0.73375	2.67887

STDEVP: standard deviation.

our planner. Thus, the dynamic ship-avoiding paths that conform to the potential field vectors can be planned.

Quantitative results. We conducted 100 Monte Carlo experiments on different path planning schemes to compare the proposed CPRRT* method with traditional methods and understand the contribution of each heuristic component to the CPRRT* performance. Figure 13 shows the dynamic environment for this experiment.

The statistical results listed in Tables 3 to 5 are consistent with the results shown in Figures 10 to 13. We captured the time of searching for each subpath once the path tree reaches the subgoal or the local planning time expires. The path searching time indicator means the sum of all subpath searching times.

The path cost and path length were obtained after the whole planning expired. If no path was returned in the given planning time, then the planning was regarded as a failure. The standard deviation (STDEVP) reflects the stability of a method.

Table 3 lists the quantitative results of planning for a virtual leader's path. The path searching time of APF is lower than 1 s, demonstrating the applicability of the radial repulsive force computation scheme. However, the planning process can be easily trapped into the local optimum. Thus, it cannot be used individually in dynamic environments.

However, the APF path is optimal in terms of the potential vectors. Thus, the smoothed APF path is utilized as the reference path for selecting subgoals. The values of path cost and path length of the APF path are regarded as the best when considering obstacles. The results of PRRT*, APF, and RRT* are considered the ground truth against which we benchmark the CPRRT* and CPRRT methods.

Although the path searching time of CPRRT* is more than 5 s, the whole planning process contains more than five local planning procedures. A local path generally contains more than 50 control steps. Thus, the waypoint update

time is probably lower than 0.1 s, which meets the control requirement, as shown by Δt in Table 2.

The path searching time of CPRRT* is slightly lower than that of PRRT*, possibly because the potential field of PRRT* does not consider the attractive potential field from the subgoals. The time STDEVP of CPRRT* is higher than that of PRRT*, mainly because CPRRT* attempts to find the OA path according to the real-time velocities, causing the large fluctuation of path searching time. However, this scenario possibly results in the superior performance indexes of CPRRT*, compared with those of PRRT*. These performance indexes include path cost, path length, path cost STDEVP, and path length STDEVP. The path length and path length STDEVP of CPRRT* are close to those of PRRT*. However, the path cost and path cost STDEVP of CPRRT* are superior to those of PRRT*. The reason is that PRRT* is guided by the pure repulsive potential field without considering the USV velocity, probably causing aggressive turns on the OA path of PRRT*, as shown in Figure 10(c).

The performance indexes of CPRRT are worse than those of CPRRT* and PRRT*. These performance indexes include path searching time, path cost, and path cost STDEVP. The main reason is that CPRRT is inclined to search for a path by extensively exploring the space rather than refining the path tree. This finding illustrates the contribution of the rewiring module of RRT*.

The abbreviation "WO" means "without". The CPRRT* without the guidance of the potential field obtains longer path searching time, higher path cost, longer path length, larger failure rate, and larger STDEVP values than those obtained by CPRRT* with potential field heuristics. This finding demonstrates that the planning process of CPRRT* without potential field heuristics becomes increasingly random and unstable, verifying the importance of the potential field guidance module.

The method without a subgoal means that the CPRRT* uses no reference path information, and it only extends the

path tree toward the global goal. The performance indexes of CPRRT* without using the reference path are much inferior to those of the method with a reference path. The reference path is planned using the given environmental information, thereby guiding the planner to extend the path in terms of historical experiences. The method without a reference path must explore space extensively to capture a path, causing long path searching time, long path, and high path cost. Its failure rate is also high because traps exist in our environment, impeding the path planning process.

However, the performance indexes of CPRRT* without potential field guidance or without a reference path are still much better than those of RRT*. RRT* explores the space randomly without assessing the probability of containing a path in a subspace, causing the exploration easy to trap in the concave obstacles.

Table 4 presents the results when CPRRT* is required to iterate for given times. The environment is shown in Figure 11. The increment of the path planning time is not linear to that of the iteration times. That is probably because the algorithm attempts to explore space extensively as the local iteration times increases, causing the path tree to be easily trapped in concave obstacles, as shown by the intensive tree branches in traps in Figure 11.

However, the increase of iterations also results in a higher path quality compared with the results of APF. As presented in Table 3, the APF results are regarded as the ground truth value of the optimal path. The path quality is rather high as the local iteration times are 10000. The observation probably demonstrates the asymptotic optimality of the CPRRT* method.

Table 5 presents the results of path planning for USVs in a formation where the environmental information is changing, as shown in Figure 13. We gain the local path searching time after a local path reaching the subgoal is found. Both the CPRRT* and PRRT* methods can meet the time requirement of control, as shown by the local path searching time and time STDEVP values. The path cost and path length values are higher than those of the virtual leader's single path in Table 3 because USVs are required to consider the collision-avoidance problem between each other. The STDEVP values are relatively small, certifying the stabilities of the methods. The paths are high-quality, as shown by the results.

The results illustrate that the CPRRT* method outperforms PRRT* in our Bug Trap environment. CPRRT* is applicable because it performs well for planning multiple paths of USVs.

Conclusion

In this article, the proposed CPRRT* method shows promising results in solving the online cooperative paths planning problem for USVs in dynamic environments. Simulations and ablation experiments were performed to illustrate the effectiveness of the heuristic modules. A

potential field was proposed by additionally considering the velocities of the USV and the DO, and the COLREGs. The performances of CPRRT* and PRRT* were compared. The experimental results verified that the new potential as well as the sampling adjustment mechanism of CPRRT* outperformed those of PRRT*. The repulsive potential force computation scheme was presented in terms of radar readings, and experiments were conducted to verify its applicability. The USVs' paths were planned by the heuristics of a formation, keeping a potential field to follow the virtual leader. Experiments were performed to certify that the method can flexibly solve the cooperative paths planning problem. Simulations were also performed in dynamic environments to confirm the online planning capability, the sea current handling capability, and the dynamic ship-avoidance ability of the CPRRT* planner.



Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported in part by the Natural Science Foundation of Liaoning Province (Grant No. 2020-MS-135), the National Natural Science Foundation of China (Grant No. 61673084), the Natural Science Foundation of Fujian Province (Grant No. 2019J01021006), the Project of Fujian Provincial Educational Research for Young and Middle-aged Teachers (Grant No. JAT190071), the Key Laboratory of Intelligent Perception and Advanced Control of State Ethnic Affairs Commission (Grant No. MD-IPAC-2019103), and Dalian High-Level Talent Innovation Support Program (Grant No. 2020RQ060).

ORCID iDs

Naifeng Wen  <https://orcid.org/0000-0003-2725-8645>
Ru-Bo Zhang  <https://orcid.org/0000-0002-6133-8078>

Supplemental material

Supplemental material for this article is available online.

References

1. Campbell S, Naeem W, and Irwin GW. A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres. *Ann Rev Control* 2012; 36: 267–283.
2. Beard RW, McLain TW, Nelson DB, et al. Decentralized cooperative aerial surveillance using fixed wing miniature UAVs. *Proc IEEE* 2006; 94: 1306–1324.
3. Lawton JRT, Beard RW, and Young BJ. A decentralized approach to formation maneuvers. *IEEE Trans Robot Automat* 2004; 19(6): 933–941.
4. Liang X, Qu X, Wang N, et al. A novel distributed and self-organized swarm control framework for underactuated

- unmanned marine vehicles. *IEEE Access* 2019; 7: 112703–112716.
5. Soulignac M. Feasible and optimal path planning in strong current fields. *IEEE Trans Robot* 2010; 27: 89–98.
 6. Liu Y and Bucknall R. Path planning algorithm for unmanned surface vehicle formations in a practical maritime environment. *Ocean Eng* 2015; 97: 126–144.
 7. Garau B, Bonet M, Alvarez A, et al. Path planning for autonomous underwater vehicles in realistic oceanic current fields: application to gliders in the western Mediterranean sea. *J Maritime Res* 2009; 6(2): 5–22.
 8. Niu H, Ji Z, Savvaris A, et al. Energy efficient path planning for unmanned surface vehicle in spatially-temporally variant environment. *Ocean Eng* 2019; 196: 106766.
 9. Wang N, Su SF, Pan XX, et al. Yaw-guided trajectory tracking control of an asymmetric underactuated surface vehicle. *IEEE Trans Ind Informat* 2018, 99: 1–1.
 10. Chakravarthy A and Ghose D. Obstacle avoidance in a dynamic environment: a collision cone approach. *IEEE Trans Syst, Man, Cyber—Part A: Syst Hum* 1998; 28(5): 562–574.
 11. Tam CK and Richard B. Collision risk assessment for ships. *J Marine Sci Technol* 2010; 15: 257–270.
 12. Statheros T, Howells G, and Maier KM. Autonomous ship collision avoidance navigation concepts, technologies and techniques. *J Navigation* 2008; 61: 129–142.
 13. Liang X, Qu X, Wang N, et al. Swarm control with collision avoidance for multiple underactuated surface vehicles. *Ocean Eng* 2019; 191: 106516.
 14. Wang N, Xie G, Pan X, et al. Full-state regulation control of asymmetric underactuated surface vehicles. *IEEE Trans Ind Electr* 2019; 66(11): 8741–8750.
 15. Wang N, Karimi HR, Li H, et al. Accurate trajectory tracking of disturbed surface vehicles: a finite-time control approach. *IEEE/ASME Trans Mechatr* 2019; 24(3): 1064–1074.
 16. Barraquand J and Latombe JC. Robot motion planning: a distributed representation approach. *Int J Robot Res* 1991; 10(6): 628–649.
 17. Lee SM, Kwon KY, and Joh J. A fuzzy logic for autonomous navigation of marine vehicle satisfying COLREG guidelines. *Int J Control Automat Syst* 2004; 2(2): 171–181.
 18. Challita U, Saad W, and Bettstetter C. Deep reinforcement learning for interference-aware path planning of cellular-connected UAVs. In: *Presented at the 2018 IEEE international conference on communications (ICC)*, Kansas City, MO, USA, 20–24 May 2018. IEEE.
 19. Ge SS and Cui YJ. New potential functions for mobile robot path planning. *IEEE Trans Robot Automat* 2000; 16(5): 615–620.
 20. Ge SS and Cui YJ. Dynamic motion planning for mobile robots using potential field method. *Autonom Robots* 2002; 13(3): 207–222.
 21. Balch T and Arkin RC. Behavior-based formation control for multirobot teams. *IEEE Trans Robot Automat* 1998; 14(6): 926–939.
 22. Loe ØAG. *Collision avoidance for unmanned surface vehicles*. Master, Norwegian University of Science and Technology, 2008.
 23. Ge SS, Cui Y, and Zhang C. Instant-goal-driven methods for behavior-based mobile robot navigation. In: *Proceedings of the 2003 IEEE international Symposium on intelligent control*, Houston, TX, USA, 8–8 October 2003. IEEE.
 24. Solari FJ, Rozenfeld AF, Sebastián VA, et al. Artificial potential fields for the obstacles avoidance system of an AUV using a mechanical scanning sonar. In: *IEEE/OES South American international symposium on oceanic engineering*, Buenos Aires, Argentina, 15–17 June 2016. IEEE.
 25. Lyu H and Yin Y. Ship's trajectory planning for collision avoidance at sea based on modified artificial potential field. In: *2017 2nd international conference on robotics and automation engineering (ICRAE)*, Dalian, China, 29–31 December 2017. IEEE.
 26. Lyu H and Yin Y. COLREGS-constrained real-time path planning for autonomous ships using modified artificial potential fields. *J Navigation* 2018; 72(3): 1–21.
 27. Wang T, Yan XP, Wang Y, et al. Ship domain model for multi-ship collision avoidance decision-making with COLREGS based on artificial potential field. *Trans Int J Marine Navigation Saf Sea Transport* 2017; 11(1): 85–92.
 28. Qin Z, Lin Z, Yang D, et al. A task-based hierarchical control strategy for autonomous motion of an unmanned surface vehicle swarm. *Appl Ocean Res* 2017; 65: 251–261.
 29. Balch T and Hybinette M. Behavior-based coordination of large-scale robot formations. In: *MultiAgent systems, 2000. Proceedings of fourth international conference on IEEE computer society*, Boston, MA, USA, 10–12 July 2000. IEEE.
 30. Balch T and Hybinette M. Social potentials for scalable multi-robot formations. In: *IEEE international conference on robotics and automation*, San Francisco, CA, USA, 24–28 April 2000. IEEE.
 31. Ge SS and Fua CH. Queues and artificial potential trenches for multirobot formations. *IEEE Trans Robot* 2005; 21(4): 646–656.
 32. Sun J, Tang J, and Lao S. Collision avoidance for cooperative UAVs With optimized artificial potential field algorithm. *IEEE Access* 2017; 5: 18382–18390.
 33. Lavelle SM and Kuffner JJ. Randomized kinodynamic planning. *Int J Robot Res* 2001; 20: 348–400.
 34. Frazzoli E. Quasi-random algorithms for real-time spacecraft motion planning and formation flight. *Acta Astronautica* 2003; 53: 485–495.
 35. Karaman S and Frazzoli E. Sampling-based algorithms for optimal motion planning. *Int J Robot Res* 2011; 30: 846–894.
 36. Islam F, Nasir J, Malik U, et al. RRT*-SMART: a rapid convergence implementation of RRT*. *Int J Adv Robot Syst* 2013; 10: 299–305.
 37. Gammell JD, Srinivasa SS, and Barfoot TD. Informed RRT*: optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. In: *IEEE/RSJ international conference on intelligent robots and systems*, Chicago, USA, 14–18 September 2014. IEEE.

38. Otte M and Emilio F. RRTX: asymptotically optimal single-query sampling-based motion planning with quick replanning. *Int J Robot Res* 2016; 35(7): S. 797–822. DOI: 10.1177/0278364915594679.
39. Wang J, Chi W, Shao M, et al. Finding a high-quality initial solution for the RRTs algorithms in 2D environments. *Robotica* 2019; 37(10): 1–18.
40. Jaillet L, Cortés J, and Siméon T. Sampling-based path planning on configuration-space costmaps. *IEEE Trans Robot* 2010; 26(4): 635–646.
41. Bruce J and Veloso MM. Real-time randomized path planning for robot navigation. In: *IEEE/RSJ international conference on intelligent robots & systems*, Lausanne, Switzerland, 30 September–5 October 2002. pp. 2383–2388. IEEE.
42. Wen N, Zhang R, Liu G, et al. Online heuristically planning for relative optimal paths using a stochastic algorithm for USVs. *J Navigation* 2019; 73(2): 1–24.
43. Qureshi AH, Mumtaz S, Iqbal KF, et al. Adaptive potential guided directional-RRT. In: *IEEE international conference on robotics & biomimetics*, Shenzhen, China, 12–14 December 2013. IEEE.
44. Qureshi AH, Iqbal KF, Qamar SM, et al. Potential guided directional-RRT* for accelerated motion planning in cluttered environments. In: *IEEE international conference on mechatronics & automation*, Takamatsu, Japan, 4–7 August 2013. IEEE.
45. Qureshi AH and Ayaz Y. Potential functions based sampling heuristic for optimal path planning. *Autonomous Robots* 2016; 40(6): 1079–1093.
46. Xinyu W, Xiaojuan L, Yong G, et al. Bidirectional potential guided RRT* for motion planning. *IEEE Access* 2019; 7(99): 95046–95057. DOI: 10.1109/ACCESS.2019.2928846.
47. Mousazadeh H and Kiapey A. Experimental evaluation of a new developed algorithm for an autonomous surface vehicle and comparison with simulink results. *China Ocean Eng* 2019; 33(3): 268–278.

Appendix

The formulas of the collision cone method are listed below.

We expand $(v_{\theta})_{PA} \cdot (v_{\theta})_{PB} \leq 0$ as follows to get a practical style of the condition: $v_i^2 \sin(\alpha - (\theta + \theta_{max})) \sin(\alpha - (\theta - \theta_{max})) + v_o^2 \sin(\beta - (\theta + \theta_{max})) \sin(\beta - (\theta - \theta_{max})) - v_i v_o [\sin(\alpha - (\theta - \theta_{max})) \sin(\beta - (\theta + \theta_{max})) + \sin(\alpha - (\theta + \theta_{max})) \sin(\beta - (\theta - \theta_{max}))] \leq 0$.

We expand the formula as: $v_i^2 (\cos(2\theta_{max}) \cos(2(\beta - \theta)) + v_o^2 (\cos(2\theta_{max}) \cos(2(\alpha - \theta)) - v_i v_o (2 \cos(\alpha - \beta) \cos(2\theta_{max}) - 2 \cos(\alpha + \beta - 2\theta))) \geq 0$.

Then, we get $v_r^2 + v_{\theta}^2 \cos(2\theta_{max}) - 2v_r^2 + v_r^2 + v_{\theta}^2 \geq 0$. Because $\cos(2\theta_{max}) = (d_{io}^2 - 2r_o^2) / d_{io}^2$, where d_{io} is the distance between USV and the obstacle, thus the condition is transformed into $d_{io}^2 v_{\theta}^2 \geq r_o^2 (v_r^2 + v_{\theta}^2)$.