



Trajectory-cell based method for the unmanned surface vehicle motion planning



Zhe Du^a, Yuanqiao Wen^{b,c,*}, Changshi Xiao^{a,b,c,**}, Liang Huang^{a,b,c}, Chunhui Zhou^{a,b,c}, Fan Zhang^{a,b,c}

^a School of Navigation, Wuhan University of Technology, Wuhan, 430063, China

^b Hubei Key Laboratory of Inland Shipping Technology, Wuhan, 430063, China

^c National Engineering Research Center for Water Transport Safety, Wuhan, 430063, China

ARTICLE INFO

Keywords:

Unmanned surface vehicle (USV)
Motion planning
Dynamic constraints
Discretization rules
Trajectory-Cell

ABSTRACT

A trajectory-cell based method was proposed for unmanned surface vehicle (USV) motion planning to combine the expression of the dynamic constraints and the discretization of the search space. The dynamic constraints were expressed by the USV trajectories produced by the mathematical model. The search space was performed by the discretization rules with the consideration of the path continuity, the search convenience and the maneuvering simplification. Therefore, the trajectory-cells were the discretized trajectories, which made the search space meet the USV dynamic constraints, and guaranteed the final spliced path continuous. After abstracting the characteristics of those cells, the available waypoints and headings were represented as the search indexes. Finally, a trajectory-cell based path searching strategy was proposed by determining the cost function of the A* algorithm. The results showed that the proposed algorithm can plan a practical motion path for the USV.

1. Introduction

Recent years, USVs have been applied in military patrolling [1–3], environmental monitoring [4,5] and maritime supervision and management [6,7]. To accomplish these missions, path planning is one of the most important premises.

According to the properties of unmanned systems, path planning is classified into three stages: route planning, trajectory planning and motion planning. Route planning is a pure geometric issue, focusing on the characteristics of the path, e.g., planning the shortest or safest path. The unmanned system in this stage is regarded as a mass point that has no kinematic and dynamic characteristics, like a pedestrian in the road map or a virtual character in the video game.

Trajectory planning is an optimization issue. Besides the path, the kinematics constraints are considered, such as speed, heading and curvature. This stage can be taken as an improvement of route planning [8], and the unmanned system refers to the fully actuated rigid body, such as industrial or humanoid robot.

Motion planning is a control issue. Unlike the first two stages, it focuses on whether the planned path can be achieved by the control system. This is because the unmanned system is usually the underactuated system (e.g., the unmanned ground vehicle (UGV) and the

unmanned aerial vehicle (UAV)), where the degree of control is less than the degree of freedom (DOF).

As to the USV, there are six-DOF (rolling, pitching, yawing, surging, swaying and heaving) and two control variables (the rudder angle and the main engine revolving speed). The inertia, resistance and response time in the water are larger than those on the ground or in the air, so the motion control of the USV is more complex than that of UGV or UAV. Moreover, for some special navigation environment, like a harbor or dock, the traffic density is higher, compared with other places that give higher request to motion control. Thus, the motion planning methods applied in the UGV or UAV domains are not suitable for the USV.

To plan a practical path for the USV, a trajectory-cell based method was proposed. The trajectory-cells not only considered all the USV dynamic constraints, but also discretized the search space to combine those constraints with search algorithm.

2. Related work

2.1. Route planning

The shortest route is a typical issue discussed at this stage [9].

* Corresponding author at: Intelligent Transportation System Research Center, Wuhan University of Technology, Wuhan, 430063, China.

** Corresponding author at: School of Navigation, Wuhan University of Technology, Wuhan, 430063, China.

E-mail addresses: wenyqwhut@foxmail.com (Y. Wen), cs_xiao@hotmail.com (C. Xiao).

Dijkstra algorithm [10] and A* algorithm [11] are the classical methods to solve the problem. The Dijkstra algorithm is used to search all the possible directions and extends outward from origination to destination. It ensures the shortest planned path, but costs much time. Hart et al. used a heuristic function [12] to guide the search direction to the destination. That is the prototype of the A* algorithm.

The USV is regarded as a mass point at the stage because of not considering dynamic constraints. The planning methods are used to combine the above classical algorithms with marine instruments or regulations. Some scholars [13] combined electronic chart (e-chart) and marine radar with the Dijkstra algorithm to achieve global and local route planning. E-chart is used to get the global environmental information. Then, with the Dijkstra algorithm, a global path is planned for the USV. During the voyage, a smoothing method is used to deal with the original radar image, which is transformed into a binary image (map) for path searching.

When it comes to collision avoidance, all the ships at sea must obey the *International Regulations for Preventing Collisions at Sea* (COLREGS). So the regulations are introduced to normalize the planned route. Some scholars [14–16] planned a global path by the A* algorithm firstly. Then, based on the COLREGS, they classified the collision into different situations, and worked out different behaviors to avoid them. Those behaviors correspond to different routes.

In summary, there are many sophisticated algorithms proposed at this stage, but the planned route is different from the real path. To make the results more real, scholars study the trajectories of the unmanned systems.

2.2. Trajectory planning

The issue at this stage focuses on curvature problem. The Dubins Path [17] is used for the problem. The main idea is that for a particle with certain speed, the shortest possible path consists of three pieces. Each piece is a straight line or an arc of a circle [18]. However, the Dubins path does not solve the continuous curvature. It is because the curvature of a straight line is zero, whereas a circle arc has a certain curvature. Moving from the straight line to the circle arc, there is a jump for the curvature from zero to a certain value. The Fermat Spiral (FS) [19] is put forward to fix it. The FS connects successive straight lines in a plane, since the curvature of this curve is equal to zero at the beginning. The other similar curves replacing the arc are the Hermite Spline curve [20], the B-Spline curve [21] and the Clothoid curve [22,23].

For the USV, the above arc or curve fitting methods can be used. The turning process is considered during the voyage, combining the “CLC” (Circle-Line-Circle) Dubins path with the genetic algorithm to solve the pose (positions and heading angles) transition from the start to the end [24]. The floating-point numbers and the turning radius of the KT equation (an equation for the ship control response) are used as the order and the radius of curvature of Bessel curve separately to restrain the path [25]. Some scholars take advantage of the multi constraint optimization to solve the trajectory planning. Kim et al. [26] considered the heading angle and the radius of curvature constraints to optimize the path. They took into account the USV possible steering angle in the planning, turning the original two-dimensional (x, y) planning space into a three-dimension (x, y and θ). Yang et al. [27] studied the angle and ship dimension constraints. The angle constraint in the LOS (Line of Sight) guide algorithm is used to improve the A* algorithm. In the collision avoidance, ship dimension and obstacle security boundary are used to make the planned path safer.

Although these methods make the planned path close to the real trajectory, the trajectory planning is an improvement of the route planning focusing on the curvature. The other constraints are ignored.

2.3. Motion planning

According to the controllability, there are two kinds of systems at this stage: the actuated (industrial robot and mechanical arm) and the underactuated (UGV, UAV, USV and AUV (Autonomous Underwater Vehicle)). For the former, all the motion poses can be controlled by the dynamic equations, which optimizes the control parameters to achieve the paths for different missions. For the latter, there are two kinds of methods. The first method includes two steps. Firstly, plan an original path by the route-planning algorithm. Then, design a controller that contains kinematics and dynamics constraints to drive the unmanned system to the destination according to the planned path [28]. The key of this method is the motion control. With different goals, the motion control can be classified into the point stabilization, the trajectory tracking and the path following [29]. Essentially, this method is used for the designed controller to reconstruct or approximate the “ideal route”.

Another method is based on the random sampling, such as the probabilistic roadmap method (PRM) and the rapidly-exploring random tree (RRT). The PRM [30] is to generate random road signs (states) in the space to determine the feasible regions (free space). Similarly, the state lattice [31] is a discretized way from the configuration space to the state space. The RRT is more applicable to the nonholonomic system (Nan, 2014), because the state transition can be improved by the kinetic model with the new node constrained by the kinetic equations [32–35].

Overall, the two methods mentioned above can be used to solve the general motion planning issues, but have their defects. The first one refers to the control theory. It focuses on the controller designing without improving the planning algorithm, so it is an indirect motion planning. The second is the improvement of the trajectory planning. Instead of considering dynamic constraints separately, these methods use the mathematical model for more constraints. However, the interactions of these constraints are still ignored.

Some scholars have some different attempts for the USV motion planning in recent years. Svec et al. [36] used the probability prediction. The USV mathematical model is used to predict all the possible trajectories. Then, min-max game-tree is used to select the smallest collision probability trajectory from the possible trajectories pool. Du et al. [37] used the USV mathematical model to build a set of trajectory segments. Through the location and direction that those segments can reach, a search strategy containing the USV movement characteristics is proposed.

According to the above analysis, the couplings of the USV dynamic constraints are strong, and the motion controller is difficult to design. To solve these problems, the work proposed a trajectory-cell based method for the USV motion planning, discretizing the search space to meet the USV dynamic constraints. Meanwhile, by the A* algorithm, the costs of distance and steering are considered. Thus, the method can be used to plan a more practical motion path for the USV. Table 1 shows the superiority of the proposed method.

3. Trajectory-cell model

3.1. Mathematical model of the USV

The maneuvering mathematical group (MMG) method [44] is used

Table 1
Superiority of trajectory-cell based method.

Method	Path Quality	Dynamic Constraints
Control theory based	Improved route planning path	Fully considered
Random sampling based	Improved trajectory planning path	Partially considered
Trajectory-cell based	Practical motion path	Fully considered

to build the USV mathematical model. The MMG modeling decomposes fluid forces and moments into several parts that affect hull, propeller and rudder. The external effects on hull are classified into the inertia and the viscosity [38]. The three-DOF (yawing (N), swaying (X) and surging (Y)) USV mathematical equation can be expressed as

$$\begin{cases} X = X_I + X_H + X_P + X_R \\ Y = Y_I + Y_H + Y_P + Y_R \\ N = N_I + N_H + N_P + N_R \end{cases} \quad (1)$$

where I , H , P and R denote the forces (or moment) of the inertia, viscosity, propeller and rudder, respectively. To simplify the problem, the effects of environment (wind, current and waves) are not considered.

(1) Inertia Model

The motion of a ship can cause disturbance of surrounding flow field, forming the fluid medium momentum (momentum) or the additional momentum (moment). By calculating the differential in the corresponding direction, the fluid inertia force and moment can be expressed as

$$\begin{cases} X_I = -(m_x \dot{u} - m_y v r - m_y \alpha_x r^2) \\ Y_I = -(m_y \dot{v} + m_x u r + m_y \alpha_x \dot{r}) \\ N_I = -[J_{zz} \dot{r} + m_y \alpha_x (\dot{v} + u r) + (m_y - m_x) u v] \end{cases} \quad (2)$$

where m_x and m_y are the additional mass along x-axis and y-axis, respectively; α_x is the coordinate along x-axis for the effect of m_y ; J_{zz} is the additional moment of inertia to z-axis; u and v are the velocity along x-axis and y-axis, respectively; r is the angular velocity of yawing.

(2) Viscosity Model

The viscosity forces and moment are related to the hull geometric features, the fluid physical properties and the ship motion states. When the first two factors are invariant, the motion states play a decisive role. By using the second-order Taylor expansion, the approximate value of the viscosity force and moment can be expressed as

$$\begin{cases} X_H = X_0 + X_u \Delta u \\ Y_H = Y_v v + Y_r r \\ N_H = N_v v + N_r r \end{cases} \quad (3)$$

where X_0 is the resistance causing by direct sailing; Δu the velocity change along x-axis; X_u , Y_v , Y_r , N_v and N_r are derivatives for hydrodynamic.

(3) Propeller Model

A propeller is the main source of a ship power. Apart from producing the thrust, it generates the lateral forces and moment. Compared to the rudder effect, the lateral force and moment can be ignored. Thus, the propeller model can be expressed as

$$\begin{cases} X_P = (1 - t_p) T \\ Y_P = 0 \\ N_P = 0 \end{cases} \quad (4)$$

where t_p is the thrust deduction coefficient; T the thrust related to the propeller diameter, the revolving speed, the water density and the speed of advance.

(4) Rudder Model

The rudder model is an important part, which produces the main control forces and moment, and determines the ship's maneuverability and trajectories. The rudder normal force can be decomposed into the

longitudinal resistance and the transverse force. The rudder model can be expressed as

$$\begin{cases} X_R = (1 - t_R) F_N \sin \delta \\ Y_R = (1 + a_H) F_N \cos \delta \\ N_R = (x_R + a_H x_H) F_N \cos \delta \end{cases} \quad (5)$$

where δ is the rudder angle; t_R the resistance deduction coefficient; a_H the correction factor; x_H the distance between the center of gravity and the point of lateral force on rudder; F_N the normal force on rudder.

3.2. Discretization rules

The discretization rules are to discretize the trajectories. According to the USV dynamic characteristics and the path search requirements, we put forward the following rules.

Rule 1: Motion state should be consistent at the start and end moments.

A feasible way for the continuous final path is to make motion state consistent at the start and end moments. For the USV, there are two requirements: (1) The rudder angle is zero; (2) the speed is identical.

The former is to keep the course stable. When the rudder angle is zero, the USV fixes on a certain course (without the influence of environment). Therefore, the transition phase between the current and the next trajectory segments will be stable. The latter is to guarantee the unchanged curvature of the adjacent segments. For a ship, when the speed and the rudder angle are fixed, the turning radius will be fixed. Thus, the curvatures of the adjacent curves keep the same in the splitting point.

Rule 2: Rudder change should be no more than once in each cell.

As the trajectory-cell is the basic unit of the final path, it should be as simple as possible. For the trajectory segment, the rudder change should not be frequent; otherwise, the final path will have inflection points. Thus, to optimize the manipulation, the rudder change should be no more than once in a cell (excluding returning the rudder angle to zero).

Rule 3: Waypoints and headings should be mapped one-to-one.

From graphic searching, the trajectory-cell provides the location and orientation (i.e., waypoint and heading) of the next node. There are three possible ways of mapping: one waypoint corresponding multi headings (Fig. 1(a)), one heading corresponding multi waypoints (Fig. 1(b)) and one waypoint corresponding one heading (one-to-one, Fig. 1(c)).

For the mapping (a), the accessible waypoints are less than the possible headings. When it comes to a waypoint with multi headings, the orientation will be the only judgment in selecting the next node, resulting in the situation of local minimum. The mapping (b) increases some waypoints to ensure that every accessible waypoint has only one heading. However, because of the redundancy points, it causes some unnecessary searches, and increases the search time. Considering the disadvantages in the first two mappings, the mapping (c) should be the optimal choice. One waypoint corresponding to one heading makes the location and orientation function in path searching, with no redundant waypoints.

Rule 4: Shape (bounding rectangle of the trajectory segment) of the cell should be a grid or based on a grid.

The shape of the cell is supposed to be diverse because of the

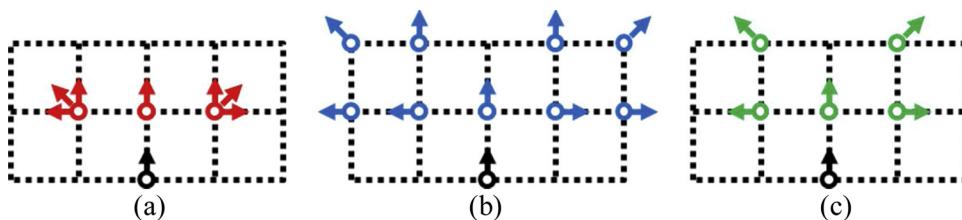


Fig. 1. Three mappings of waypoints and headings: (a) One waypoint corresponding multi headings; (b) One heading corresponding multi waypoints; (c) One waypoint corresponding one heading.

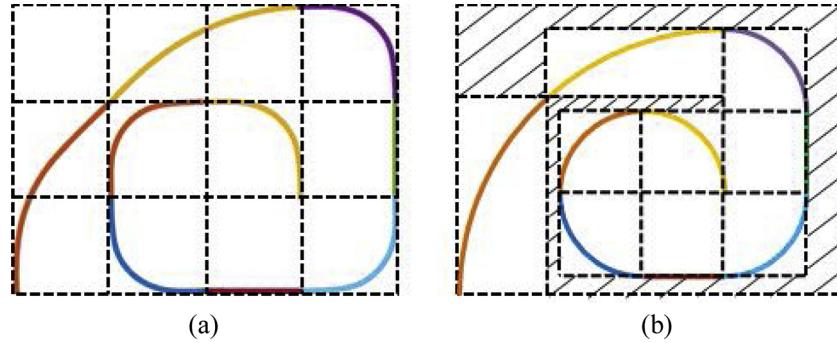


Fig. 2. Two situations of space coverage: (a) complete coverage with one basic grid cell; (b) incomplete coverage with different grid cell.

different trajectories. Considering the uniformity (convenient to search) of the space discretizing, the shape of the cell should be grid. When the size (the number of grids that a trajectory takes) of the trajectory segment is larger than that of one grid, the shape should be based on a grid (See Fig. 2(a)). Otherwise, when these trajectory segments are spliced, the space cannot be completely covered (In Fig. 2(b), the shaded area is the space not covered by the trajectory). This results in one map having several scale standards, not good for locating and waypoint searching.

Therefore, **Rule 1** standardizes the trajectory; **Rule 2** optimizes the manipulation; **Rule 3** optimizes the search; **Rule 4** standardizes the space.

3.3. Process of establishing trajectory-cell (TC)

The traditional path search algorithms concern the eight surrounding nodes in a grid map, which can be reached directly (See Fig. 3(a)). However, for the USV, the reachable waypoints are determined by the original heading. According to the change of heading, the eight original headings can be classified into two types: *Type 1*, the horizontal-vertical orientations (heading for 0, 90, 180 and 270°, and the true north is 0°); *Type 2*, the sidelings 45-degree orientations (heading for 45, 135, 225 and 315°).

(1) TC of Type 1

When the original heading is 0°, the headings 135, 180 and 225° are the opposite direction, and the nodes *d*, *e*, *f*, *g* and *h* are hardly reached through one steering. So the rest nodes (*a*, *b* and *c* in Fig. 3(b)) and headings (0, 45, 90, 270 and 315°) become the candidates. Node *b* is right in front of the original point *o*, so the heading of *b* is 0°. Nodes *a* and *c* are located on the port and starboard of the original point *o*, respectively, so the headings are 270 and 90°. Until now, the eight surrounding nodes have been chosen or excluded. To ensure the location and orientation of the node conforming to the USV dynamic constraints, the headings 45 and 315° are bound by the nodes *j* and *i*, respectively (See Fig. 3(c)). Thus, the mathematical model of producing TCs is as follows.

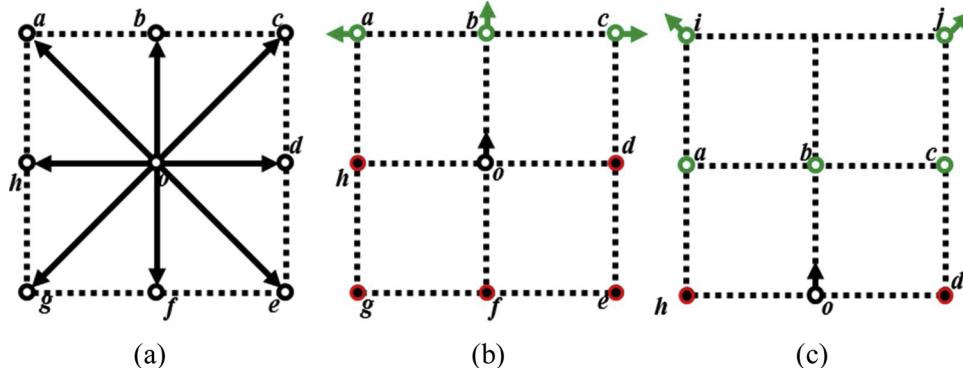


Fig. 3. Surrounding nodes that can be reached: (a) is the situation for tradition path search; (b) and (c) are the situation for the USV.

Assuming that the original state of an USV is S_{start} ($x_0, y_0, \psi_0, v_0, \delta_0$), and final state is S_{end} ($x_t, y_t, \psi_t, v_t, \delta_t$), the model of the TC is

$$TC = \{S_{start} \rightarrow S_{end}\} \quad (6)$$

$$\text{s. t. } \begin{cases} v_0 = v_t \\ \delta_0 = \delta_t = 0^\circ \end{cases} \quad (7)$$

$$\begin{aligned} TC_0 &: \begin{cases} oa: x_t = x_0 - a, \quad y_t = y_0 + a, \quad \psi_t = \psi_0 + 270^\circ \\ ob: x_t = x_0, \quad y_t = y_0 + a, \quad \psi_t = \psi_0 \\ oc: x_t = x_0 + a, \quad y_t = y_0 + a, \quad \psi_t = \psi_0 + 90^\circ \\ oi: x_t = x_0 - a, \quad y_t = y_0 + 2a, \quad \psi_t = \psi_0 + 315^\circ \\ oj: x_t = x_0 + a, \quad y_t = y_0 + 2a, \quad \psi_t = \psi_0 + 45^\circ \end{cases} \end{aligned} \quad (8)$$

where x_i and y_i are the coordinates of the position; ψ_i is the heading; v_i the speed; δ_i the rudder angle; a the side length of the grid. Eq. (7) is the premise for generating the TC, and Eq. (8) shows the five different waypoints and headings of TC_0 .

Fig. 4 shows the process of producing TC_0 . TC_0ob has no steering, so the trajectory is a straight keeping heading 0°. TC_0oa , TC_0oc , TC_0oi and TC_0oj have changed their original headings, so they have to make a steering. As the headings TC_0oa and TC_0oc changed 90°, and those of TC_0oi and TC_0oj changed 45°, the rudder angle of the former (δ_1) is larger than the later (δ_2). It is worth noting that there is a manipulating of returning rudder in each TC_0 (See Fig. 4, the red point). This is to guarantee the rudder angle is 0° when the USV arrives to the waypoints. Through the whole process, it can be found as follows:

- 1) The speed and rudder angle of the original and final state are the same, and the values of rudder angle are 0. It follows the **rule 1**.
- 2) There is only one steering in the whole process of generating each TC, which follows the **rule 2**.
- 3) For each waypoint, there is one and only one corresponding heading, which follows the **rule 3**.
- 4) The shape of the basic TC is a grid (TC_0oa and TC_0oc) and the other extensional TC is based on a grid (TC_0oi and TC_0oj), which follows

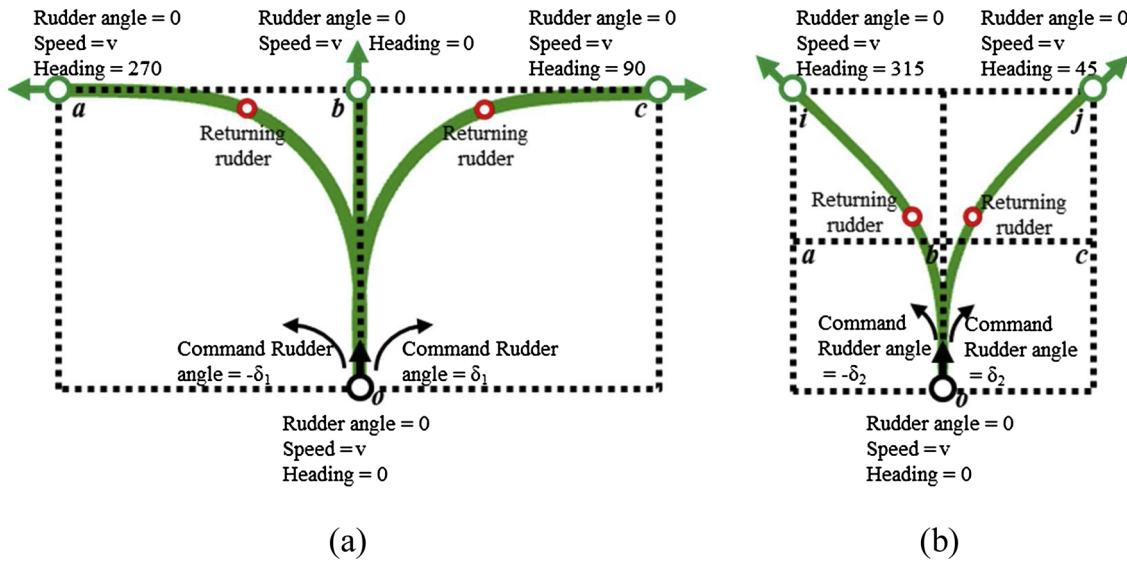


Fig. 4. Processes of producing TC₀: (a) TC_{0oa}, TC_{0ob} and TC_{0oc}; (b) TC_{0oi} and TC_{0oj}.

the rule 4.

Thus, the produced TC₀ conforms to the four rules.

(2) TC of Type 2

Type 2 can be regarded as a transition from Type 1. When the original heading is 45°, the final headings are 0, 45 and 90° (See Fig. 5(a)). From 45 to 45°, the waypoint should be node c without heading change. From 45 to 0 and 90°, the waypoints should be j and k, respectively. The mathematical model of the TC₄₅ is

$$\begin{cases} oj: x_t = x_0 + a, \quad y_t = y_0 + 2a, \quad \psi_t = \psi_0 - 45^\circ \\ oc: x_t = x_0 + a, \quad y_t = y_0 + a, \quad \psi_t = \psi_0 \\ ok: x_t = x_0 + 2a, \quad y_t = y_0 + a, \quad \psi_t = \psi_0 + 45^\circ \end{cases} \quad (9)$$

The process of producing TC₄₅ is similar to TC₀. TC_{45oc} is the straight keeping the original heading. Both TC_{45oj} and TC_{45ok} have the manipulating of steering and rudder returning. The more details can be found in Fig. 5(b), and the produced TC₄₅ conforms to the four rules.

3.4. TC manipulating table

Based on the discussion above, a complete set of TCs can be produced. In order to provide a reference of control order for the USV, Table 2 and 3 shows the TC Manipulating Table.

4. Motion planning algorithm

The searching algorithm is the core of the work. This chapter focuses on the strategy of TC based motion planning.

The algorithm is to summarize the property of those TCs as the constraints of the waypoints and headings. Thus, the USV dynamic constraints are transformed to those points and headings in the path searching, and the motion planning problem is transformed to the graph searching problem. Section 4.1 shows the process of the abstraction. Section 4.2 introduces the strategy of waypoints chosen by the A* algorithm. Section 4.3 discusses some necessary matters in dealing with obstacles. Section 4.4 is the summary and gives the algorithm flow.

4.1. Trajectory-cell abstracting

Fig. 6(a) shows the trajectories of the eight directions. After abstracting the waypoints and headings, Fig. 6(b) shows the reachable points and the corresponding headings.

There are 16 waypoints classified into two types: the inner points (the green ones) and the outer points (the blue ones). The inner points are the traditional eight surrounding nodes, and each point has its own heading. The outer points are located on a layer of outside, and the

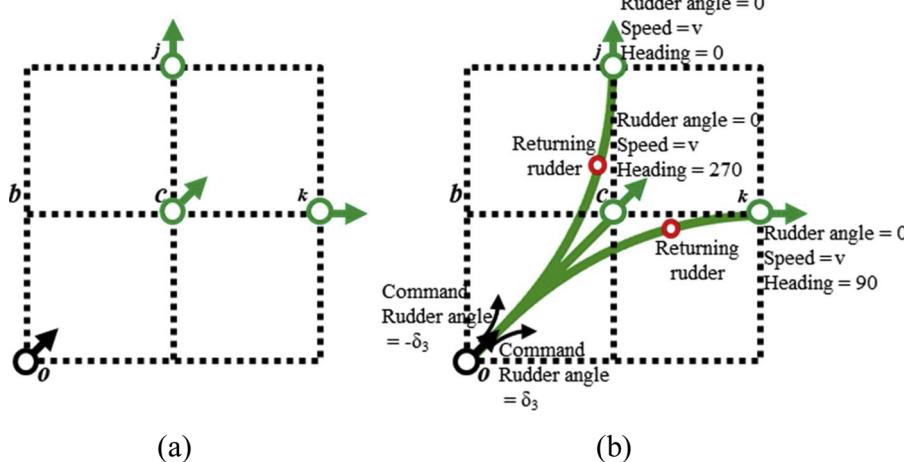


Fig. 5. Model of TC₄₅: (a) Waypoints and headings; (b) Processes of producing.

Table 2
TC Manipulating Table 1^a.

Trajectory-Cell					
Original State	H = 0° RA = 0° CRA = 0°	H = 0° RA = 0 CRA = -25°; if H > 252° RRZ	H = 0° RA = 0 CRA = 22°; if H > 74° RRZ	H = 315° RA = 0° CRA = 0°	H = 45° RA = 0° CRA = 0°
Steering Command					
Final State	H = 0° RA = 0°	H = 270° RA = 0°	H = 90° RA = 0°	H = 315° RA = 0°	H = 45° RA = 0°

Table 3
TC Manipulating Table 2^a.

Trajectory-Cell				
Original State	H = 0° RA = 0°	H = 0° RA = 0	H = 45° RA = 0°	H = 45° RA = 0°
Steering Command	CRA = -9°; if H > 218°	CRA = 8°; if H > 39°	CRA = 8°; if H < 5°	CRA = 5°; if H > 89°
Final State	H = 315° RA = 0°	H = 45° RA = 0°	H = 0° RA = 0°	H = 90° RA = 0°

^a H stands for the Heading; RA the rudder angle (positive value is starboard helm, and negative value is port helm); CRA the command rudder angle; RRZ the rudder returning to zero.

heading change from original point (ψ_0) to those points is 45° (i.e., 0° → 45°, 45° → 0°, and 45° → 90°). Thus, the 16 waypoints should be searched in one loop.

4.2. Waypoint choosing

Heuristic algorithm is often used for path planning, because it is fast, effective, simple, practicable and easy to modify [39]. The A* algorithm is the most typical one. The core of A* algorithm is the cost function $f(x)$.

$$f(x) = g(x) + h(x) \quad (10)$$

where $g(x)$ is the actual cost from start node to current node; $h(x)$ the heuristic cost (or estimated cost) from current node to final node. The smaller value of $f(x)$ leads to the better waypoint. Thus, the A* algorithm is used to choose the waypoint.

4.2.1. Actual cost

Considering the symmetry, all the TCs can be classified into five kinds of curves: 0° → 0°, 0° → 45°, 0° → 90°, 45° → 45° and 45° → 0° (See the five different colors in Fig. 6(a)). Moreover, the curve 45° → 0° can be regarded as an inverse process of 0° → 45°. Therefore, the two TCs can be treated as one kind. Eventually, the basic elements of the motion planning consist of four different TCs (See the four different color lines Fig. 7(a)).

For the USV, both distance and steering can be taken as the judgment for the quality of the path (the less the distance and steering, the better the path), so the actual cost consists of distance cost and steering cost.

(1) Distance Cost

The distance cost is a common evaluation index in path planning. It determines the distance of the final path, which is important for time and fuel saving.

TC 0° → 0° and 45° → 45° are the segments of straight line. The value is the Euclidean distance between the two points.

$$dv_S(\psi_0 \rightarrow \psi_t) = \sqrt{(x_t - x_0)^2 + (y_t - y_0)^2} \quad (11)$$

TC 0° → 45° and 0° → 90° are curve segments. The value is estimated by fitting arc. Curve 0° → 90° (See the yellow one in Fig. 7 (a)) takes up one grid, with the fitting arc as a quarter circle and the radius as the side length of grid (See the yellow dotted arc in Fig. 7 (a)). Curve 0° → 45° takes up two grids, and the fitting arc should be a part of a circle whose center is out of the grid. After calculation, when the radius is 2.5 times side length of grid, the arc can fit the curve. Thus, the distance value of curves can be expressed as

$$dv_C(\psi_0 \rightarrow \psi_t) = \frac{\alpha\pi}{180} r \quad (12)$$

where α is the central angle, and r the radius of fitting arc.

However, the mentioned distances are just the values. For the cost, they still need to be processed. Fig. 7(b) shows the number of track segments to reach the same destination. From point A to B, the blue needs 6 segments; the yellow and purple need 3; the red needs 2. That means 6 blue TCs, 3 yellow TCs, 3 purple TCs and 2 red TCs have the same results. Thus, the distance costs of these four TCs can be expressed as

$$d(\psi_0 \rightarrow \psi_t) = dv(\psi_0 \rightarrow \psi_t) \cdot a_D \quad (13)$$

where dv is the actual distance value (including d_{S} and d_{C}); a_D the distance coefficient.

(2) Steering Cost

The steering cost determines the complexity of manipulating, which is important for motion control. The change between original and final headings can reflect the cost of steering. The larger of the change is, the higher the steering cost. So it can be expressed as

$$t(\psi_0 \rightarrow \psi_t) = |\psi_t - \psi_0| \cdot a_T \quad (14)$$

where ψ_0 and ψ_t are the initial and final headings, respectively; a_T is the steering coefficient.

(3) Cost Standardization

The distance cost and steering cost make up the actual cost, but the measure units of the two costs are different. Before calculating, it needs standardization. By the Min-Max method [40], the two costs can be standardized as

$$D(\psi_0 \rightarrow \psi_t) = \frac{d(\psi_0 \rightarrow \psi_t)}{d_{max}} \quad (15)$$

$$T(\psi_0 \rightarrow \psi_t) = \frac{t(\psi_0 \rightarrow \psi_t)}{\Delta\psi_{max}}$$

where d_{max} is the maximum distance; $\Delta\psi_{max}$ the maximum heading change.

Therefore, the actual cost is

$$g(\psi_0 \rightarrow \psi_t) = D(\psi_0 \rightarrow \psi_t) \cdot \omega + T(\psi_0 \rightarrow \psi_t) \cdot 1 - \omega \quad (16)$$

where ω is the weight coefficient.

4.2.2. Heuristic cost

Heuristic cost estimates the future path, assessing which waypoint is the most potential for the destination. Therefore, this cost is to induce

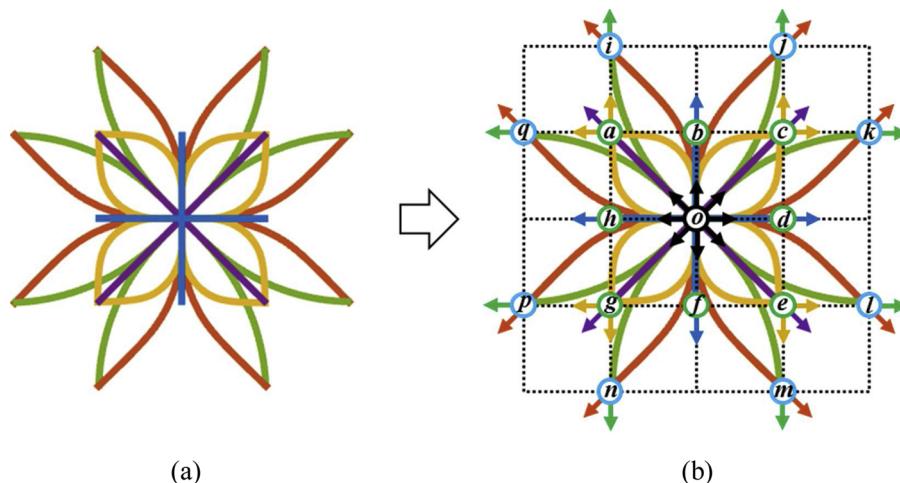


Fig. 6. Situations of eight directions: (a) Trajectories of TCs; (b) Extracted waypoints and headings (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

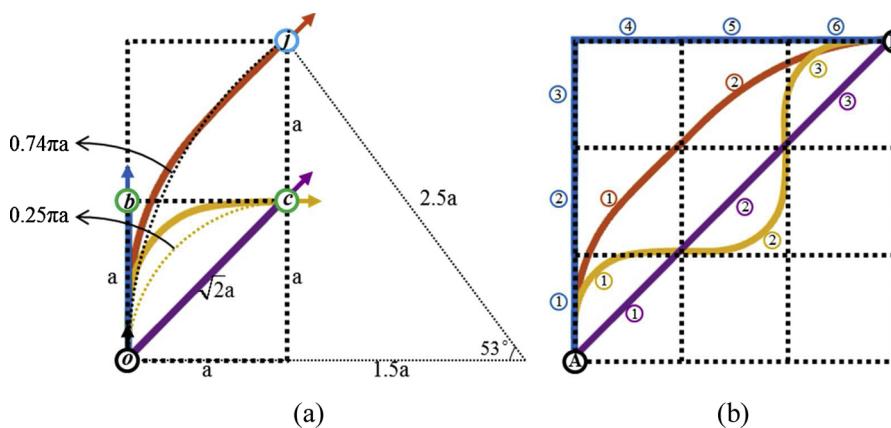


Fig. 7. Actual cost: (a) Actual distance. Blue line segment is $0^\circ \rightarrow 0^\circ$; purple line segment $45^\circ \rightarrow 45^\circ$; yellow curve segment $0^\circ \rightarrow 90^\circ$; red curve segment $0^\circ \rightarrow 45^\circ$. (b) Number of segments for the four TCs reaching the same destination (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

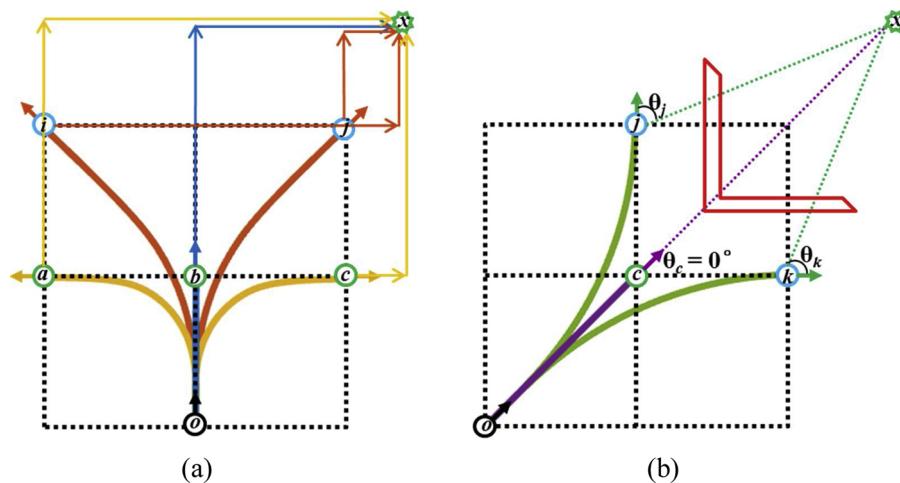


Fig. 8. Heuristic cost: (a) Manhattan distance; (b) Using heading as the judgment.

the searching direction toward the destination. The Manhattan distance is used here.

Due to the **Rule 3**, the different headings have different Manhattan distance values. In addition, the smaller the value is, the closer the waypoint is to the goal. For example, in Fig. 8(a), waypoint a (270°), b (0°), c (90°), i (315°) and j (45°) have different Manhattan distances, and j is the node closest to the goal x . Thus, the heuristic cost can be expressed as

$$h(\psi_0 \rightarrow \psi_t) = |x_g - x_t| + |y_g - y_t| \quad (17)$$

where x_g and y_g are the coordinates of the goal position.

In fact, the heading can be used for a reference. A variable θ is defined as the angle from the heading to the direction of the goal. The smaller θ means the closer heading to the direction of the goal. Especially, when $\theta = 0^\circ$, the heading is toward the goal; when $\theta = 180^\circ$, the heading is opposite to the goal. However, if there are obstacles in the search space, the closest heading may not be optimal. In Fig. 8(b),

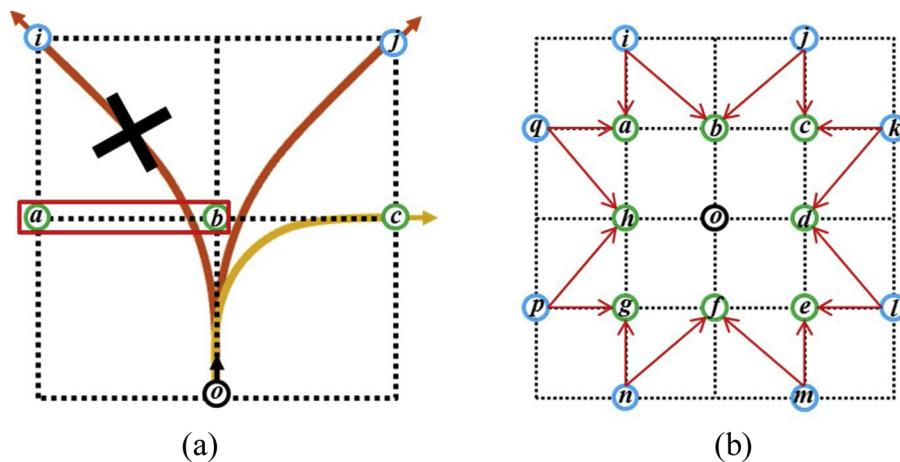


Fig. 9. Correlation of waypoints: (a) special situation of unreachable; (b) correlation of the 16 waypoints.

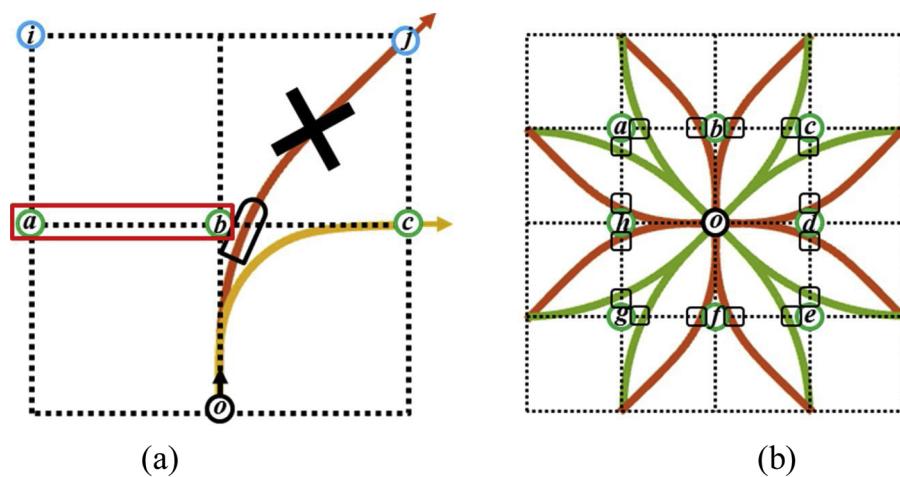


Fig. 10. Dimension of the USV: (a) Special situation considering dimension constraint; (b) Proximity points (the black box) of the TCs.

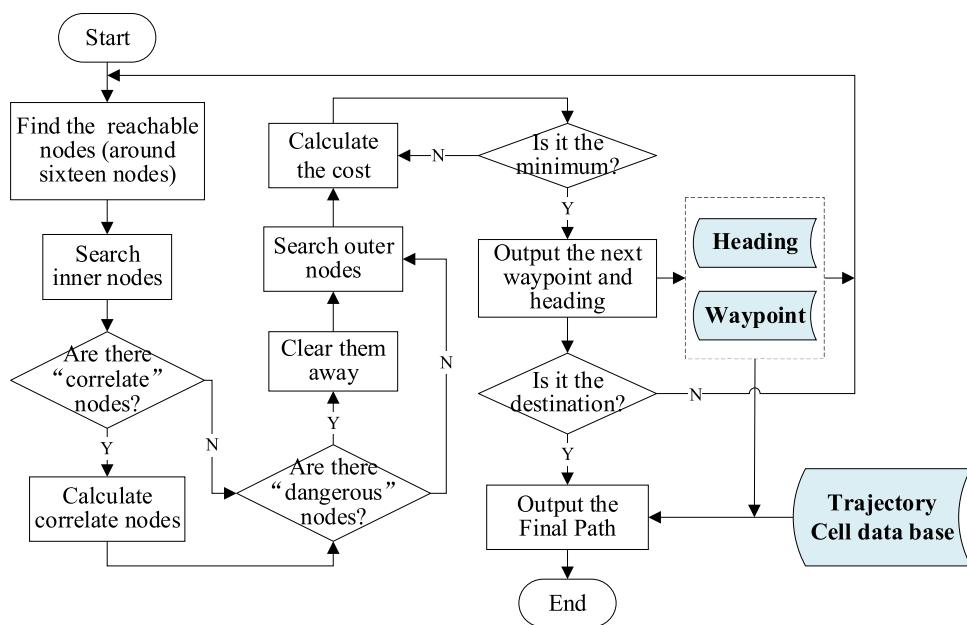


Fig. 11. Flow of the algorithm.

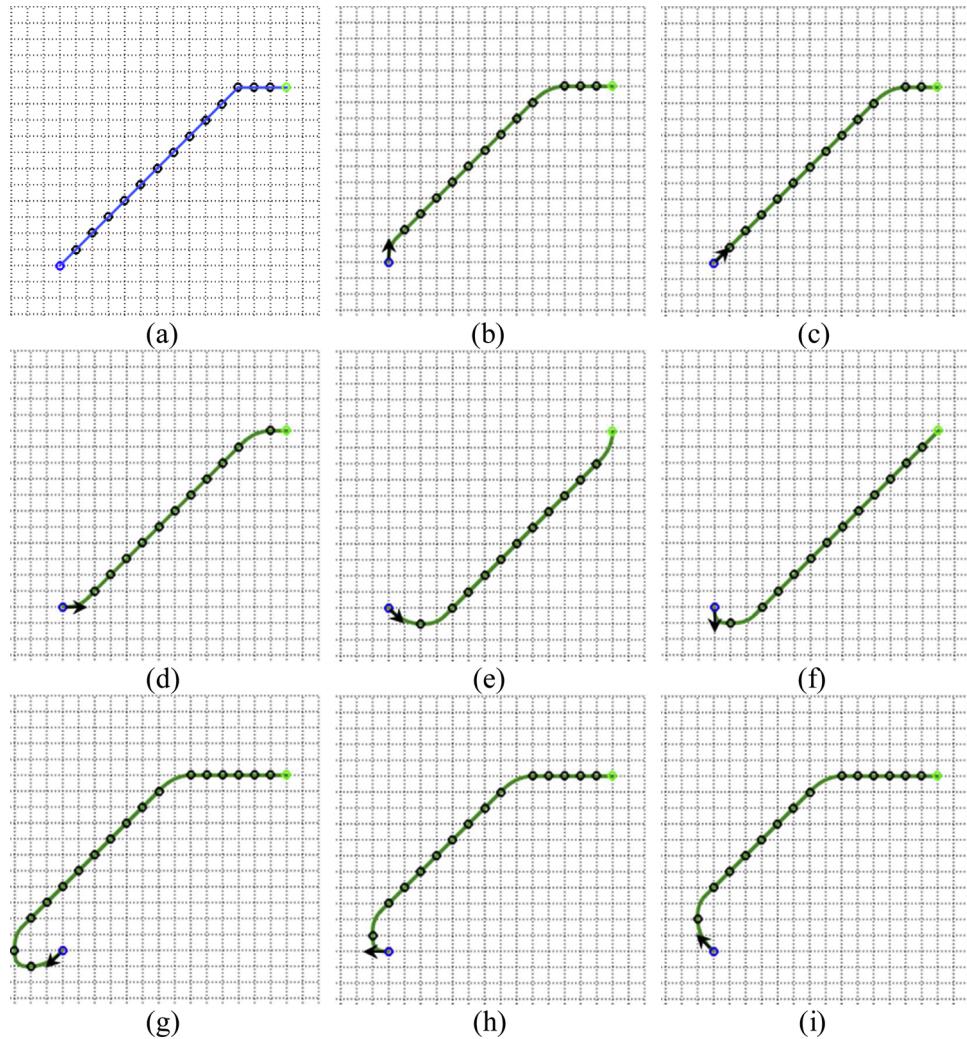


Fig. 12. Planning results in free space. (a) Route planning. (b)–(i) Motion planning with different original headings: (b) 0° , (c) 45° , (d) 90° , (e) 135° , (f) 180° , (g) 225° , (h) 270° , (i) 315° .

waypoint c is the best choice from heading, because $\theta_c = 0^\circ$. In fact, there is an obstacle (the red one) between waypoint c and goal x . To clear the obstacle, the waypoint should be chosen from j or k . Thus, it is not appropriate to take the heading as a reference of the heuristic cost.

4.3. Strategy of obstacle avoidance

4.3.1. Waypoints correlating

When it comes to the obstacles, the relationship between inner and outer points becomes important.

Fig. 9 (a) shows the waypoints a and b are unreachable, with accessible i, j and c . As a and b are occupied with obstacles, the way from o to i is blocked. That means waypoint i cannot be reached. Thus, it is necessary to correlate the inner points and the outer points.

Fig. 9 (b) shows the correlation of the whole 16 waypoints. Each outer point has two related inner points. As long as the two inner points are unreachable, the correlated outer point cannot be reached.

4.3.2. The constraint of ship dimension

Apart from dynamic constraints, the ship dimension should be considered. Fig. 9 (a) shows trajectory oj is close to the waypoint b . If considering dimension, the USV can crash into the obstacle (See Fig. 10 (a)). Thus, the trajectories of TC should be analyzed.

The curvature of $TC\ 0^\circ \rightarrow 0^\circ$ and $45^\circ \rightarrow 45^\circ$ is 0, with no “dangerous” point. The curvature of $TC\ 0^\circ \rightarrow 90^\circ$ is large, with the curve away from

the “dangerous” point. The curvature of $TC\ 0^\circ \rightarrow 45^\circ$ is small, and the heading changes slowly, so this curve has the chance to encounter the “dangerous” point. Fig. 10 (b) shows the situations of the “dangerous” point, marked with the black box.

4.4. Flow of the algorithm

Fig. 11 shows the flowchart of the algorithm based on trajectory-cell model, A* search, waypoints correlating and ship dimension constraint.

At the beginning of the algorithm, the sixteen nodes around the start are searched for finding the reachable nodes according to the initial waypoint and heading. Then, the inner nodes are taken into account for judging whether there are “correlate” nodes. If it is so, calculate and find them; otherwise, continue to make a judgment whether there are “dangerous” nodes. After excluding all the “dangerous” nodes, the outer nodes are put into the candidate pool along with the previous inner points. Next, the costs (actual and heuristic) of the candidates are calculated, and the one with minimum value is selected as the next node. At the same time, the waypoint and the heading of the selected node are saved. If the next node is not the destination, the waypoint and the heading are taken as the index for the next searching loop; otherwise the searching loop is ended. Finally, the planned path is reconstructed inversely according to the series of waypoints and headings saved previously and the trajectory-cell database.

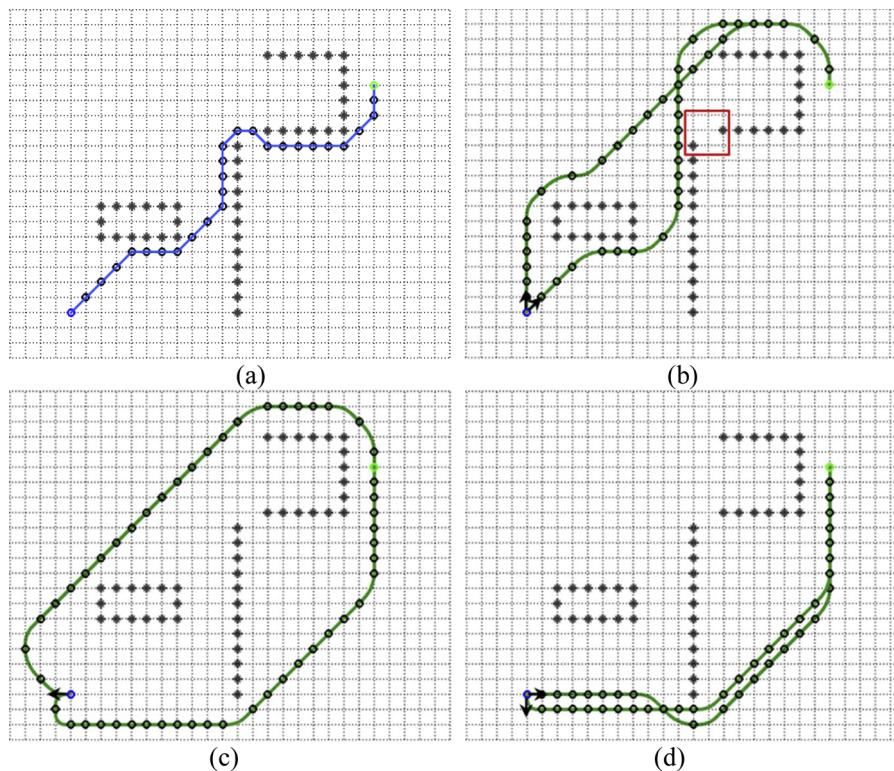


Fig. 13. Planning results in obstacle space. (a) Route planning. (b)-(d) Motion planning with different original headings: (b) 0 and 45°, (c) 270°, (d) 90° and 180°.

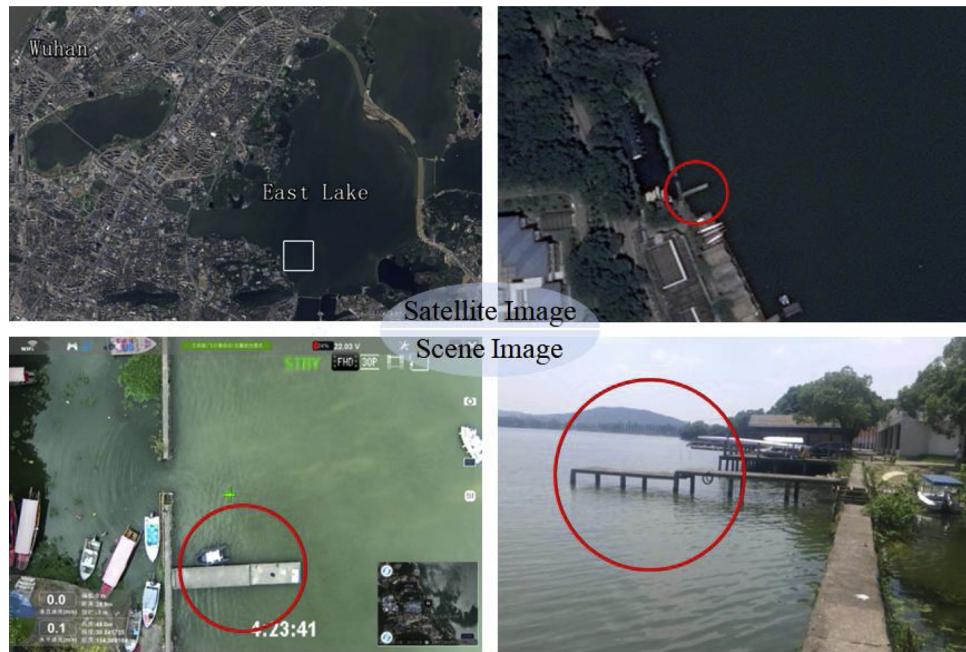


Fig. 14. Satellite and scene images of the trial site: East Lake in Wuhan, China.

5. Experiments and results

This Chapter is to validate the algorithm. The simulation experiment compares the route planning and the motion planning, and the field experiment compares the planned path and the real trajectory.

5.1. Simulation experiment

In the same conditions, the simulation experiment is divided into

two parts.

(1) In Free Space

Fig. 12 shows the results in free space. The blue point is the start; the green one the end; the black one the waypoint.

Fig. 12 (a) is the result of the route planning. It is a shortest path from the start to the end. **Figs. 12** (b)-(i) are the results of the motion planning. Besides considering the length of the path, there is an adjustment based on their states in the results. Thus, the experiment shows that in the premise of comforting the USV dynamic constraints,

iNav-II					
Length	Breadth	Draught	Displacement		
3.96 m	1.55 m	0.3 ~ 0.5 m	0.708 t		
Rudder Angle			Ship Speed on Different Gears after Steady		
Max	Min	I	II	III	IV
25°	-25°	2.2 m/s	2.0 m/s	1.55 m/s	1.0 m/s
Sensor					
1. DGPS (RTK)	2. Anemorumbometer	3. Inertial Measurement Unit (IMU)	4. Camera	5. Millimeter-Wave Radar	

Fig. 15. Detail of the experimental USV.

the proposed algorithm can plan a relatively optimal path in free space.

(2) In Obstacle Space

Fig. 13 shows the results in obstacle space. To make the results more contrastive, the layout of the obstacles is relatively complex.

Fig. 13 (a) shows the result of the route planning. Although the space between those obstacles is narrow, it still finds a shortest path without collision. Figs. 13 (b)-(d) show the results of the motion planning. The original headings of Figure (b) are 45 and 0°. Compared to Figure (a), the motion path of 45° does not go through the interspace between the strip-type and the concave obstacles (the red box). This is because the interspace is too narrow to meet the constraints of the USV dynamic and the ship dimension. The motion path of 0° chooses a different way in the beginning, because it is to adapt to the initial heading.

Fig. 13 (c) shows the situations of original heading 270°, and there are two results: the paths of the below and the above. The below path is relatively short, but there is a sharp turning at the beginning. The above path is longer, but the manipulating is better. Both are reasonable just for the different requirements. Fig. 13 (d) shows the situations of original heading 90 and 180°. The two results are similar because the two original headings are easy to bypass the strip-type obstacle from the below without unnecessary steering.

When it comes to obstacles, the collision avoidance becomes the first priority. On the premise of the safety, the route planning still focuses on the shortest path, while the motion planning emphasizes the maneuverability and the dimension of the USV.

5.2. Field experiment

5.2.1. Trial site and the USV

The field experiment is conducted in East Lake in Wuhan, China. Fig. 14 shows the satellite and scene images. There is a small dock (the red circle) stretching out of the lakeshore, which is the start point.

As for the USV, it is a second-generation experimental boat named “iNav-II” which is a glass steel boat propelled by electric. Fig. 15 shows the detail. It is equipped with some necessary sensors, like DGPS (RTK, centimeter-level accuracy) for locating position, Inertial Measurement Unit (IMU) for measuring pose, anemorumbometer for detecting the direction and speed of wind, millimeter-wave radar for detecting obstacles and camera for acquiring real-time image. The data and steering instruction are exchanged by radio communication system.

The experimental area is a square of about 600 m × 600 m. As the ship domain is a safety zone for protecting a ship from other around ships, the size of basic TC in this trial is determined based on the USV Domain. According to some researches [41–43], the ship domain is usually an ellipse whose major axis is 4 to 8 times the length of the ship. Five times is adopted in the work. The size of a basic TC is 4 m × 5 = 20 m (See Fig. 16). Thus, there are 30 × 30 grids in the experimental area.

5.2.2. Results

The propeller gear is in a position of III in all the field experiments. Therefore, in the experiment, the average speed of the USV is about 1.55 m/s. There are three sets of experiments, and in each set, there are four tests from simplicity to complexity.

Fig. 17 shows the first experiment, and the set (a) is the planned path. The scenarios in (a1) is the simplest, with straight planned path. The scenarios in (a2) and (a3) are similar, but the planned paths are different because of the shape of obstacle. The scenario in (a4) is more complex because the cove of the two “L”-shaped obstacles is toward the outside. The USV is easy to be trapped in this situation, but the planned path bypasses them to the destination.

The sets (b) and (c) are the real path and the distance error, respectively. As there is no steering, the real path (b1) conforms to the (a1), with the errors no more than 5 m in the process. The results in real paths (b2), (b3) and (b4) are not as good as (b1), and the errors are up to 10 m. However, all the values of the error in this experiment are no more than the size value of a basic trajectory-cell. Considering the maneuverability of the USV and the environment influence (especially the wind) in the field, the results are acceptable.

Fig. 18 shows the second experiment. The destination is located on the north-northeast of the dock, so there is a steering on port side in the planned path (a1). The tests (2) to (4) successively increase one obstacle. The planned paths (a2) to (a4) adjust from different directions to the destination. Especially, no planned path is trapped into the U-shaped or T-shaped obstacle.



Fig. 16. Experimental area and the size of a basic TC (based on the length of the USV).

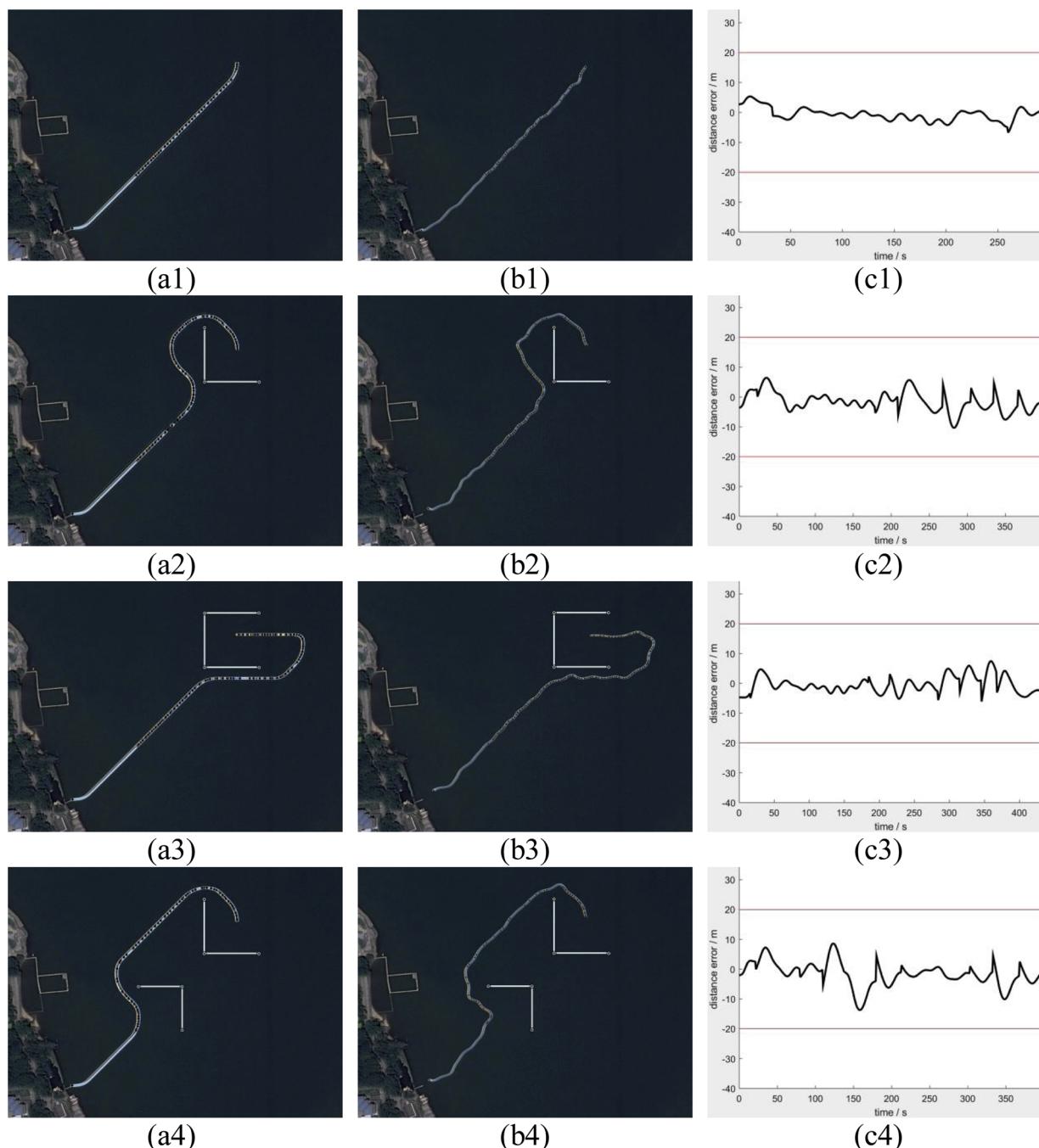


Fig. 17. The first experiment. Index (a) is the planned path, (b) the real path, (c) the distance error between the two paths. Indexes (1)–(4) are the four different tests.

Compared to the real paths, the value of distance error in the test (1) is still the lowest (no more than 5 m). The values in the tests (2) to (4) are higher, and the maximum appears on the steering around, such as 250–300 s in the test (2) and 0–150 s and 300–400 s in the test (3). From an overall perspective, the errors between the real path and the planned path are acceptable.

Fig. 19 shows the third experiment. Except for the personal error at the beginning of test 4 (the origination), all the values of the error in this trial are still no more than the size value of a basic trajectory-cell.

In addition, Fig. 20 shows the speed of the USV in the most three

complex tests (Figs. 17 (b4), 18(b4) and 19(b4)). When the motion of the USV is stable, the speed is fixed on 1.55 m/s as the nominal speed on propeller gear III (See Fig. 15). It can verify that the start and end moments of the speed in each trajectory-cell can be kept consistent (**Rule 1**).

The field experiments prove the feasibility of the proposed method. The comparison of the planned paths and the real paths shows that despite the influence of the real-time circumstance, the real paths conform to the planned path. All the values of the distance error are no more than the size value of a basic trajectory-cell.

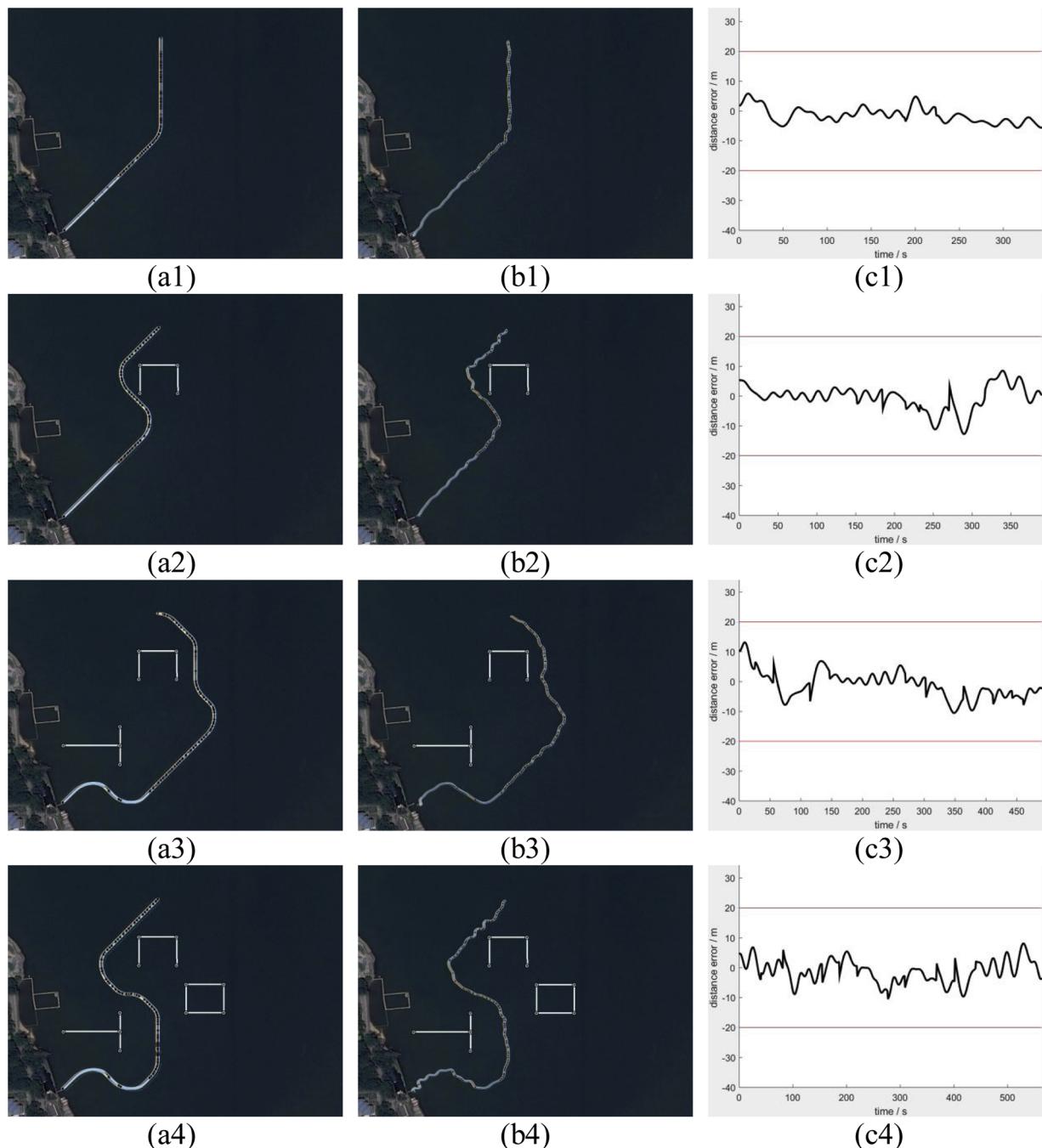


Fig. 18. The second experiment. Index (a) is the planned path; (b) the real path; (c) the distance error between the two paths. Indexes (1)–(4) are the four different tests.

6. Conclusions

The work proposed a method based on trajectory-cell for the USV motion planning. The trajectory-cell is a set of trajectory segments for combining the USV trajectories and the discretization rules. The trajectories are produced by the USV mathematical model, which can express the dynamic constraints.

The discretization rules are used to divide a long trajectory into several trajectory segments with certain characteristics. Thus, the trajectory-cell can discretize the search space into many subspaces properly and make each one match the USV dynamic constraints.

Meanwhile, the trajectory-cell guarantees the continuous final path when those segments are spliced. Based on the trajectory-cell model,

the motion planning algorithm is proposed by taking the waypoints and headings as the search indexes and the A* algorithm as the search strategy. The algorithm not only meets the USV dynamic constraints, but also considers the cost of distance and manipulation and the constraint of ship dimension. The simulation and field experimental results indicate that the proposed algorithm can plan a more practical motion path for the USV.

However, the work can be further improved. As shown in the field experiment, the real paths are affected by the environment. To make the algorithm more robust in practice and get a better result, it is necessary to add the environmental disturbance model (wind, wave and current) into the USV mathematical model.

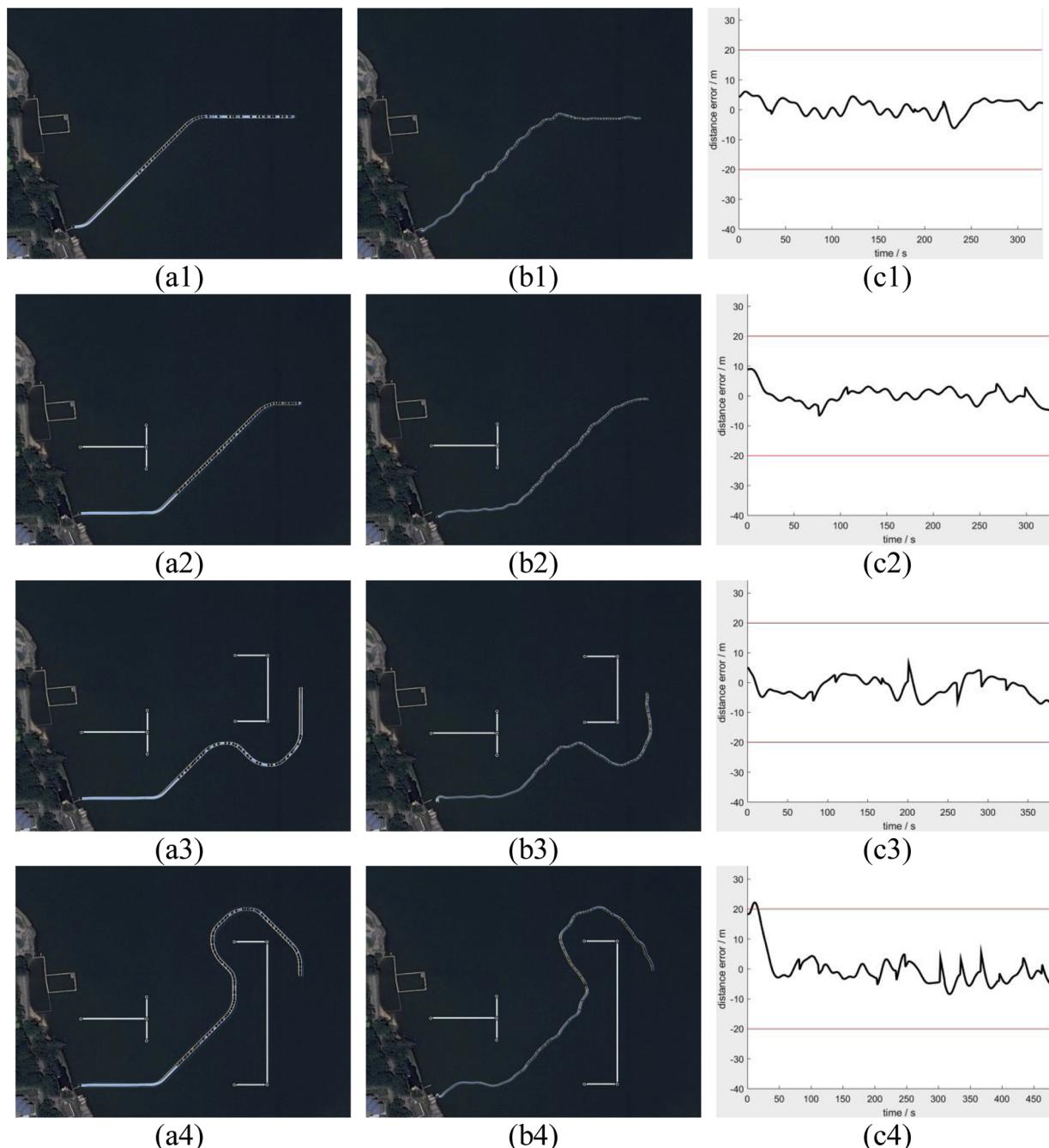


Fig. 19. The third experiment. Index (a) is the planned path; (b) the real path; (c) the distance error between the two paths. Indexes (1)-(4) are the four different tests.

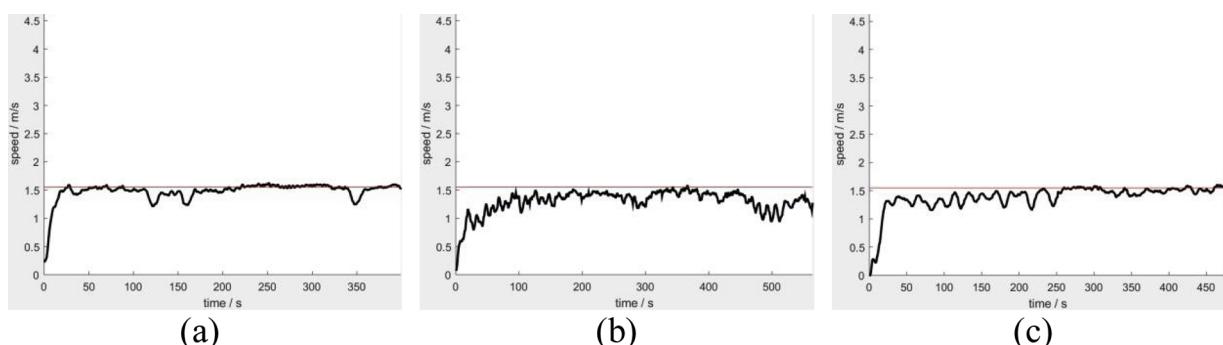


Fig. 20. Speed of USV in typical trials. (a) is the test in Fig. 17 (b4); (b) the test in Fig. 18 (b4); (c) the test in Fig. 19 (b4).

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) (Grant No. 51679180, No. 51579204 and No. 51709218), Double First-rate Project of WUT and the National Key R&D Program of China (Grant No. 2018YFC1407405 and No. 2018YFC0213904).

References

- [1] V. Bertram, Unmanned surface vehicles-a survey, *Skibsteknisk Selskab* (2008) 1–4.
- [2] Y. Niu, C. Shen, B. Dai, X. Xu, et al., The development of unmanned battle system, *Chin. Natl. Defense Technol.* 30 (5) (2009) 1–11.
- [3] R.J. Yan, S. Pang, Development and missions of unmanned surface vehicle, *J. Mar. Sci. Appl.* 9 (4) (2010) 451–457.
- [4] J. Koo, S. Jung, H. Myung, A jellyfish distribution management system using an unmanned aerial vehicle and unmanned surface vehicles, *Underwater Technology (UT)*, IEEE, 2017, pp. 1–5.
- [5] G. Hitz, F. Pomerlesau, F. Colas, et al., State estimation for shore monitoring using an autonomous surface vessel, *Experimental Robotics*, Springer International Publishing, 2016, pp. 745–760.
- [6] M. Caccia, M. Bibuli, R. Bono, et al., Unmanned surface vehicle for coastal and protected waters applications: the charlie project, *Mar. Technol. Soc. J.* 41 (2) (2007) 62–71 (10).
- [7] J. Zhang, J. Xiong, G. Zhang, et al., Flooding disaster oriented USV & UAV system development & demonstration, *Oceans*, IEEE, 2016, pp. 1–4.
- [8] C. Christophe, G. Zoltan, S. Benjamin, et al., Mobile robot trajectory planning under kinematic and dynamic constraints for partial and full field coverage, *J. Field Robot.* 34 (7) (2017) 1297–1312.
- [9] K. Song, C. Chang, Navigation integration of a mobile robot in dynamic environments, *J. Field Robot.* 16 (7) (1999) 387–404.
- [10] N.C. Guo, A solution to best itinerary problem based on strategy set under Dijkstra algorithm, *Appl. Mech. Mater.* 333 (2013) 1442–1445.
- [11] J. Peng, Y. Huang, G. Luo, Robot path planning based on improved a* algorithm, *Cybern. Inf. Technol.* 15 (2) (2015) 171–180.
- [12] P. Hart, N. Nilsson, B. Raphael, A formal basis for the heuristic determination of minimum cost paths, *IEEE Trans. Syst. Sci. Cybern.* 4 (2) (1968) 100–107.
- [13] J. Zhuang, L. Wan, Y. Liao, et al., Global path planning of unmanned surface vehicle based on electronic chart, *Comput. Sci.* 38 (09) (2006) 211–214 219.
- [14] Y. Lu, Research on Path Planning Algorithm of Unmanned Surface Vehicle, Harbin Engineering University, 2010.
- [15] H. Yang, Research on Path Planning Algorithm of USV, Dalian Maritime University, 2016.
- [16] S. Campbell, A rule-based heuristic method for COLREGS-compliant collision avoidance for an unmanned surface vehicle, *Manoeuvring and Control of Marine Craft*, (2012), pp. 386–391.
- [17] F. Dougherty, G. Woolweaver, At-sea testing of an unmanned underwater vehicle flight control system, *Proceedings of the 1990 Symposium on Autonomous Underwater Vehicle Technology* (1990) 65–73.
- [18] R.P. Anderson, D. Milutinovic, A stochastic approach to dubins vehicle tracking problems, *Transactions on Automatic Control* 59 (10) (2014) 2801–2806.
- [19] A.R. Dahl, Path Planning and Guidance for Marine Surface Vessels, Norwegian University of Science and Technology, 2013.
- [20] F.N. Fritsch, R.E. Carlson, Monotone piecewise cubic interpolation, *SIAM J. Numer. Anal.* 17 (2) (1980) 238–246.
- [21] H. Li, W. Wang, K. Li, A parallel parking path plan based on the B spline theory, *J. China Highway* 09 (2016) 143–151.
- [22] Y. Wang, X. Zhu, Z. Zhou, A method of UAV path planning based on the Clothoid curve, *J. Northwestern Polytech. Univ.* 06 (2012) 874–878.
- [23] S. Madhavan, T. Antonios, W. Brian, et al., Cooperative path planning of multiple UAVs using Dubins paths with clothoid arcs, *Control Eng. Pract.* 18 (9) (2010) 1084–1092.
- [24] Y. Chen, A Preliminary Study on the Optimized Collaborative Strategy and Implementation of the Unmanned Surface Vehicle, Jiangsu University of Science and Technology, 2016.
- [25] X. Sun, Research on the Real-time Path Planning System of Unmanned Surface Vehicle, Dalian Maritime University, 2016.
- [26] H. Kim, D. Kim, J.U. Shin, et al., Angular rate-constrained path planning algorithm for unmanned surface vehicles, *Ocean. Eng.* 84 (4) (2014) 37–44.
- [27] J.M. Yang, C.M. Tseng, P.S. Tseng, Path planning on satellite images for unmanned surface vehicles, *Int. Joural Nav. Archit. Ocean. Eng.* 7 (1) (2015) 87–99.
- [28] L. Kang, Several Algorithmic Researches on Motion Planning for Autonomous Mobile Robot, Nanjing University of Science and Technology, 2010.
- [29] H. Yu, Nonholonomic Wheeled Mobile Robot Motion Planning and Control Research, Ocean University of China, 2014.
- [30] L. Kavraki, J. Latombe, Randomized preprocessing of configuration space for fast path planning, *IEEE International Conference Robotics and Automation*, (1994), pp. 2138–2139.
- [31] M. Pivtoraiko, A. Kelly, Generating near minimal spanning control sets for constrained motion planning in discrete state spaces, *The Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)* (2005) 3231–3237.
- [32] N. Xu, X. Chen, Q. Kong, et al., Motion planning for robot with nonholonomic constraints, *ROBOT* 06 (2011) 666–672.
- [33] M. Du, T. Mei, J. Chen, et al., RRT-based motion planning algorithm for intelligent vehicle in complex environments, *Robot* 04 (2015) 443–450.
- [34] J. Song, B. Dai, E. Shan, An improved RRT path planning algorithm, *Acta Electron. Sin.* 38 (2010) 225–228.
- [35] D. Belter, P. Labocki, S. Piotr, Adaptive motion planning for autonomous rough terrain traversal with a walking robot, *J. Field Robot.* 33 (3) (2016) 337–370.
- [36] P. Svec, A. Thakur, B.C. Shah, et al., USV trajectory planning for time varying motion goals in an environment with obstacles, *ASME 2012 International Design Engineering Technical Conferences (IDETC) & Computers and Information in Engineering Conference (CIE)*, (2012), pp. 1297–1306.
- [37] Z. Du, Y. Wen, C. Xiao, et al., Motion planning for unmanned surface vehicle based on trajectory unit, *Ocean. Eng.* 151 (151) (2018) 46–56.
- [38] K. Kijima, T. Katssuno, Y. Nakiri, et al., On the manoeuvring performance of a ship with the parameter of loading condition, *The Japan Shipbuilding Society Proceedings*, (1990), p. 168.
- [39] C. Reeves, *Modern Heuristic Techniques for Combinatorial Problems*, Blackwell Scientific Publications, Oxford, 1993.
- [40] G. Yajun, Y. Pingtao, Properties analysis for method of linear dimensionless, *J. Stat. Res. Iran* 02 (2008) 93–100.
- [41] Y. Fujii, K. Tanaka, Traffic capacity, *J. Navig.* 24 (4) (1971) 543–552.
- [42] H. Chen, G. Guo, Research of ship domain and the traffic capacity in paratactic bridge water area, *Ship Ocean Eng.* 37 (5) (2008) 113–116.
- [43] M.G. Hansen, T.K. Jensen, S.T. Lehn, et al., Empirical ship domain based on AIS data, *J. Navig.* 66 (6) (2013) 931–940.
- [44] Oyama, Ogawa, MMG Report-I-Mathematical Model of Control Movement, The Japan Shipbuilding Society, 1997.