

Toward Time-Optimal Trajectory Planning for Autonomous Ship Maneuvering in Close-Range Encounters

Guoyuan Li, *Member, IEEE*, Hans Petter Hildre,
and Houxiang Zhang, *Senior Member, IEEE*

Abstract

Ship intelligence has been a hot topic in recent years. How to achieve autonomous maneuvers in a complex marine environment in a safe, efficient, and low-cost manner is a fundamental task that ocean engineers face. This paper presents a two-stage trajectory planning scheme to address the minimum-time maneuvering problem in close-range encounters. The scheme is robust and versatile, as it can deal with the complex spatial variability, such as sea current, state constraints, marine traffic, and physical constraints, of close-range maneuvering. In the first stage, a directed graph with variable length is generated according to the sea current distribution. A wavefront search is applied on the graph to explore the reachability, the cost of state constraints, and the risk of collision. After a discrete solution has been found, the second stage involves searching for a smooth solution. A Bézier curve based parameter optimization approach is proposed to get rid of limited moving directions in the directed graph and explore around the discrete path. The result will be a near-optimal, smooth path. The proposed scheme has been tested to solve the Zermelo's ship steering problem and several other close-range maneuvering problems. The results demonstrate that the scheme is efficient in generating smoothed minimum-time trajectories for surface vessels when maneuvering in close-range encounters.

Index Terms

Ship intelligence, trajectory planning, sea current, wavefront search, collision avoidance, Bézier curve.

G. Li, H. P. Hildre and H. Zhang are with the Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, Ålesund, 6025, Norway (e-mail: guoyuan.li@ntnu.no).

NOMENCLATURE

x, y, ϕ	Ship position and orientation.
v, u	Ship surge speed and total speed.
θ, δ	The thrust direction of the ship and the maximum angular velocity of the thrust.
t_f	The travel time to reach the final state.
c, α	Sea current velocity and direction.
$\Delta c, \Delta \alpha$	The changes of velocity magnitude and angle of sea current.
$\Delta l, L$	Node's interval and its maximum connection length.
P	The state of the own ship.
\mathcal{S}, \mathcal{D}	The set of the static and dynamic obstacles.
D_0, D_s, D_d	The dimension of the own ship, the static obstacle and the dynamic obstacle.
Q_0, Q_s, Q_d	Vertex format of D_0, D_s and D_d .
C_s, C_d	The array of clearance distances of the static and dynamic obstacles.
X_s, X_d	The position of the static and dynamic obstacles.
V_d, Φ_d	The velocity and orientation of the dynamic obstacles.
U_0, U_{t_f}	The cost function for initial and final states.
T_0	The time constant used in U_0 .
C_{t_f}	The radius defined in U_{t_f} .
τ, λ	Position control parameters for the connection point and the control point.

I. INTRODUCTION

Interest in developing advanced vessels that have intelligence and are capable of executing different levels of autonomy for maritime operations has increased in recent years. The term “levels of autonomy” is often used to describe the degree to which the vessel can act on its own. Autonomy can range from a vessel being completely controlled by human, to being fully autonomous and without any interaction from humans. In some cases, subtasks of a maritime operation can be fully autonomous. For example, ship navigation in the open sea can be nearly fully autonomous, while passing narrow waters is more likely to require close human supervision and decision making, or even full human-operation.

To pave the way for autonomous ships, economic, regulatory and technological factors need to be brought together to build up the requisite level of ship intelligence. The autonomy of modern marine vessels is steadily increasing through strongly interacting subsystems, including trajectory planning, motion prediction, and thrust allocation. These subsystems may be dedicated to a specific, primary objective of the vessel or be part of the general essential ship operations. Because it is challenging to achieve full autonomy for general maritime operations, the industry has been focusing on developing intelligent subsystems for specific maritime operations, such as the autonomous docking system from Rolls-Royce Marine AS [1] and Wärtsilä [2], and the AUTOSEA project by Kongsberg Maritime [3]. This paper focuses on the maneuver subsystem in line with that trend.

As the basis of autonomous ships, how to generate effective trajectories is of great importance, especially for complex maneuvering scenarios. Recently, the maritime industry increasingly demands surface vessels that can maneuver in close-range operating areas, such as dynamic positioning for wind turbine installation, towing boats in narrow channels, and ship-ship docking for loading/unloading. The limited working space, the positioning, and the heading requirements for operations and the marine traffic nearby constitute a complex spatial environment. In addition, the presence of uncertainty, especially the environmental perturbations like wind, wave, and sea current, increases maneuvering complexity [4].

In such a context, planning an optimal trajectory before an autonomous maneuver in the close-range operating area is necessary, which increases ship safety to some extent. Indeed, path planning has generated extensive interest for years [5]. The primary goal of path planning is to find a sequence of way points that connect the initial and the final configuration under

certain physical constraints like acceleration limitation, while optimizing goals, e.g., searching for energy-efficient [6], time-optimal [7] or shortest paths [8].

In general, solving a planning problem requires defining and constructing a countable state space and conducting efficient searches according to certain criteria [5]. Work on this subject dates back to the 1970s. Notable results include the potential field method [9] and the elastic band method [10]. However, these methods often reach a local minimum and are not easy to extend to deal with complex constraints. Later work proposed sampling-based planning [11], [12], a method using certain heuristics to discover and improve collision-free paths. These algorithms can satisfy complex constraints but take exponential search time. As a branch of path planning, graph-based search approaches such as the Dijkstra's algorithm and the A^* algorithm have proven efficient for solving path planning problems in a low-dimensional space [13], [14]. However, the smoothness of the graph-based solutions depends on the discretized level of the state space. Coarse discretization improves the computational efficiency but results in insufficiently smooth paths. Therefore, researchers have sought smoothing technologies, such as splines, polynomials, and the Bézier curve [15], [16]. There are also lattice-based planners that use control model primitives to search in state space [17], [18]. Such planners are suitable for complex scenarios, but have a relatively low re-planning rate due to computational complexity. To reduce the computation time, a preferred solution is to use hierarchical approaches [18], where a high-level planner, such as the designed planner in this paper, runs at a low re-planning frequency to search for the optimal trajectory within the entire domain, followed by a high-frequency low-level planner to deal with the local changes in the environment [19]. For those path planning problems that can be represented as a set of parameters, evolutionary methods including genetic algorithm (GA) [20] and particle swarm optimization [21], plus reinforcement learning [22], are applicable. Recent work has proposed ways of utilizing optimization technologies for path planning based on proper assumptions, including the gradient decent method [23], the stochastic optimization [24], the dynamic programming principle [25], and the optimal control method [26], [27], [28].

Regarding trajectory optimization for marine vehicles, the constant need to increase economic feasibility under the premise of the vehicle's safety motivates various implementations of vehicle routing systems [29]. Minimizing the travel time for vehicles that are subject to weather constraint has attracted particular interest. Investigations have generally assumed that vehicles use constant engine power as they sail, while varying their headings to follow a time-optimal path. Both discrete and continuous solutions are available. For instance, Soullignac et al. proposed a Dijkstra-

like algorithm using symbolic wavefront expansion to search both the path and the departure time for minimizing the travel time and verified the method in the presence of a time-varying flow field [30]. Zeng et al. utilized a B-spline-based quantum particle swarm optimization technique to minimize travel time while maintaining simultaneous arrival time for all the participating vehicles [31]. Rhoads et al. characterized the minimal time problem of the fixed speed autonomous underwater vehicles as a Hamilton-Jacobi equation, which can be solved using the extremal field method [32]. Lolla et al. did similar work, except that they utilized the level set approach to repeatedly compute the reachable region until the goal point is involved [33].

Collision avoidance is not emphasized in the papers listed above since the marine vehicles are assumed to be operating in open sea; however, for vehicles in close-range operating areas, collision avoidance becomes paramount, especially for encounters where the own ship (OS) is required to give way to the target ship (TS) according to the collision regulations (COLREGs) [34]. In fact, there have been attempts to create COLREGs compliant path planning algorithms for surface vehicles in dynamic marine traffic [35], [36], [37], [38]. As these works aim to explore collision-free paths, environmental conditions are often simplified or ignored. But trajectory planning for autonomous maneuvering in close-range operating area cannot utilize any off-the-shelf methods, due to its complex spatial variability.

Our ongoing project aims to develop intelligent digital twins of autonomous ships to provide life cycle services, ranging from risk assessment, trajectory planning, prediction, and trajectory tracking to force allocation. The present work focuses on trajectory planning and proposes a variant of the Dijkstra's algorithm to address the close-range maneuvering problem. The most significant finding is that:

- The algorithm can generate a near time-optimal path for surface vehicles with nonholonomic motion constraints. Unlike the traditional Dijkstra based methods used for ground vehicles, this algorithm is specially designed by taking the dynamic marine traffic, the COLREGs, and the ocean current effects into account.
- Smoothing the obtained path while keeping it near-optimal is the second contribution of this paper. Specifically, we propose a new approach to connect Bézier curves with C^2 continuity. Representing the obtained path as parameters of a group of Bézier curves and optimizing through GA under the same criteria makes it possible to generate a similar but smoothed path.

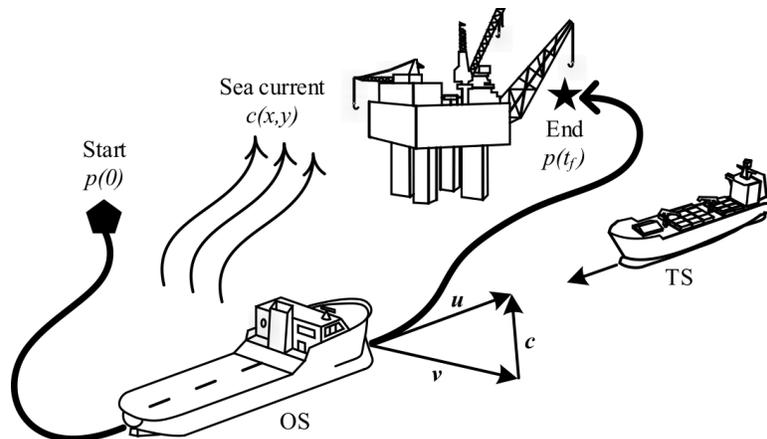


Fig. 1. Sketch of time-optimal path planning for surface vehicles in close-range maneuvering mission (c : sea current, v : vehicle velocity in calm water and u : vehicle velocity over the ground).

II. PROBLEM FORMULATION

This section describes the problem of time-optimal planning of surface vehicles with both interior and exterior constraints in close-range operating areas, together with an optimization framework to address the problem.

A. Close-range Maneuvering

Maneuvering surface vehicles for close-range maritime operation, e.g., steering the ship towards an oil platform for loading goods, as depicted in Fig. 1, needs to not only guarantee ship safety but also ensure that there is enough time and space for the subsequent operations. In particular, the following aspects are of great concern in the mission:

- **Environmental disturbances:** Wave, wind, and sea current are the main factors affecting ship motion. However, missions are often executed in calm weather when wave and wind are negligible. Thus the sea current is the dominant environmental impact considered here [4]. Moreover, sea current is assumed to be time-invariant in the task, as changes occur on the order of days [39].
- **Thrust limitation:** The ship has limited engine power, which means it cannot completely compensate for the effects of the sea current on the ship. Therefore, it is critical to take advantage of the environmental impact during the maneuvering. Furthermore, the changes of the thrust direction are restricted to the physical properties of active thrusters, e.g., the

maximum angular velocity of an azimuth thruster. Therefore, the corresponding steering ability should be identified [27].

- **Collision avoidance:** There are both static and dynamic obstacles in the mission. The difficulty is to navigate among them according to COLREGs, since TSs may not be able, or may choose not, to comply with the rules [37]. In addition, ship dimensions should be taken into consideration for close-range maneuvering. A clearance constraint by expanding the obstacles virtually is one of the possible ways to prevent collision [28].
- **Surge velocity:** Low moving speed over the ground is suggested, such that the pilot has enough time to respond to avoid collision. To this end, it is often assumed that the magnitude of ship velocity in calm water is no greater than that of maximum velocity of the sea current [26], [33].
- **Ship orientation:** A specific task will typically require a vehicle to have a particular orientation, especially its heading at the end position [40]. This applies to the close-range maneuvering task when the ship is approaching the oil rig. Ship heading is also an important element used for determining encounter types and assessing risk of collision [41]. It therefore plays a key role in decision making in the whole maneuvering process.

B. Surface Vehicle Kinematic Model and Constraints

Considering the low surge speed requirement and the thrust limitation during close-range maneuvering, the kinematics of a surface vehicle is expressed as

$$\begin{cases} \dot{x} = v \cos(\theta) + c \cos(\alpha) \\ \dot{y} = v \sin(\theta) + c \sin(\alpha) \end{cases} \quad (1)$$

where $X = (x, y)$ is the position of the vehicle; $v > 0$ is the magnitude of vehicle velocity in calm water; θ stands for the thrust orientation in world frame; $c > 0$ denotes the velocity magnitude of the sea current; and α represents the sea current orientation in world frame. The angular velocity of thrust orientation is constrained by:

$$|\dot{\theta}| \leq \delta \quad (2)$$

where δ refers to the maximum angular velocity. We assume $v < \max(c)$; otherwise the vehicle can move freely. Note sideslip is omitted here and the vehicle moves along the tangent direction

of the path, as illustrated in Fig. 1. Thus, the vehicle's total velocity and orientation angle is determined by

$$\begin{cases} u = \sqrt{\dot{x}^2 + \dot{y}^2} \\ \phi = \arctan \frac{\dot{y}}{\dot{x}} \end{cases} \quad (3)$$

Let $P(t) := (X, \theta, \phi)$ be the state of the vehicle (OS) and t_f be the time of reaching the final state. In order to achieve collision avoidance, we assume the set \mathcal{S} representing the static obstacles' positions and dimensions, and the set \mathcal{D} denoting the TSs' initial states including the position, the velocity, the orientation and the dimension, are available. Given the OS's dimension D_0 , and initial and final states P_0 and P_{t_f} , the minimum time maneuvering problem becomes:

$$\begin{aligned} & \min_{\theta} t_f \\ & \text{s.t.} \begin{cases} \text{Eq. (1) (2)} \\ 0 < Col(P(t), D_0, \mathcal{S}) < \infty \\ 0 < Col(P(t), D_0, \mathcal{S}) < \infty \\ P(0) = P_0 \\ P(t_f) = P_{t_f} \\ t \in [0, t_f] \end{cases} \end{aligned} \quad (4)$$

wherein $Col(\cdot)$ is the function of collision risk assessment, with values less than infinity indicating safe maneuver amongst obstacles.

Remark 1: Complete initial/final states may make the problem unsolvable. Relaxing the constraints by involving only partial initial/final states, such as X , (X, ϕ) , or (X, θ) , renders a solution to (4) possible [40].

C. Two-stage Trajectory Generation Scheme

To address the maneuvering planning problem listed in (4), a two-stage trajectory planning scheme is proposed. An overall flowchart of this approach is illustrated in Fig. 2. The first stage aims to use the wavefront search to find a discrete path on a grid representing the maneuvering working space. For each wavefront node, under the premise of reachability constrained by (1) and (2), its cost is evaluated by the arrival time from the initial state, in conjunction with the collision risk assessment via COLREGs. As far as the wavefront search reaches the final state, a discrete solution is found.

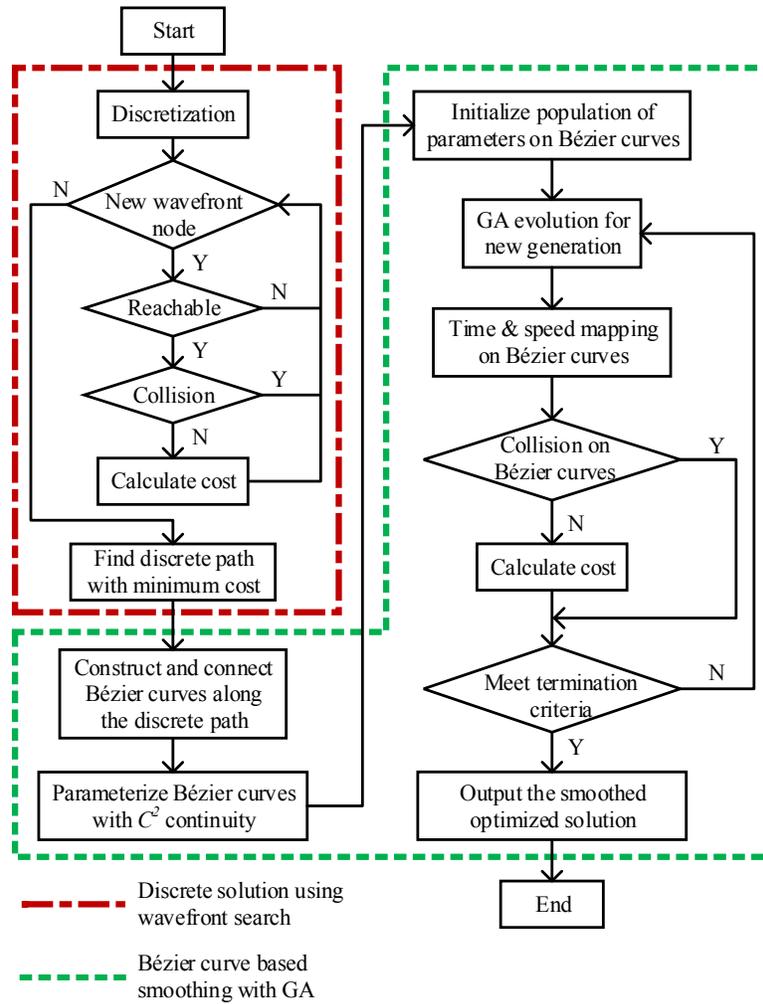


Fig. 2. Flowchart describing the two-stage scheme for close-range maneuver planning problems.

Traditional graph-based search approaches usually suffer from limited motion directions. For example, in [30], only the nearest eight neighboring nodes can be accessed. Although refining the grid and enabling the adaptive connection between nodes, as introduced in Section III-A, are beneficial to increase possible moving directions to some degrees, a further optimization that is immune to motion direction restriction is needed. This is achieved in the second stage, which attempts to search the space near the discrete solution to obtain a near-optimal, smooth result. The grid constraint is eliminated by replacing the discrete path with parameterized Bézier curves with C^2 continuity (see Section IV-A for details). Thus, the optimization turns into searching in the parameter space of the Bézier curves, without considering any grid constraints. GA makes it possible to obtain the optimized parameters constructing the Bézier curves, which consequently

form a smooth path.

III. WAVEFRONT SEARCH IN DYNAMIC ENVIRONMENT

The related work in Section I reveals that searching in the state space can start either from the initial state to the final state as with the Dijkstra's algorithm, or vice versa, as with the Hamilton-Jacobi method. The former formulates the wavefront search strategy, which explores the outer boundary of the attainable region. By contrast the latter evaluates the value function of the state backward in time, from which a control law can be generated along the negative gradient of the value function. The advantage of the backward search technology is that it provides a continuous solution. However, the approximation of the value function in the state space is computationally expensive [26]. Furthermore, although the backward search can find solutions in a domain among static obstacles [40], it is not applicable to dynamic obstacles. This is because the backward search lacks time information from the initial state to the current state to estimate the state of dynamic objects. Therefore, a wavefront search is preferable in the two-stage scheme.

This section introduces the key elements involved in the first stage of the scheme in Fig. 2, including how to establish a directed graph with variable connections, judge node reachability, and comply with COLREGs and user-defined initial/final states—to complete the wavefront search algorithm.

A. Directed Graph Generation

Let the working space for close-range maneuvering be uniformly divided into $m \times n$ nodes, with an interval of Δl . The generation of the directed graph is based on the assumption that the sea current vector on one node is similar to its connected neighbors, so that the sea current on these nodes is considered consistent when the vehicle moves from the node to one of its neighbors.

The similarity of sea current is defined as follows: For two non-zero vectors of sea current \mathbf{c}_i and \mathbf{c}_j , as shown in Fig. 3a, \mathbf{c}_j is *similar* to \mathbf{c}_i only if their difference with respect to percentage of magnitude is no greater than Δc , and their angle difference is no greater than $\Delta \alpha$. That is,

$$\begin{cases} \frac{||\mathbf{c}_i|| - ||\mathbf{c}_j||}{||\mathbf{c}_i||} \leq \Delta c \\ \arccos\left(\frac{\mathbf{c}_i \cdot \mathbf{c}_j}{||\mathbf{c}_i|| ||\mathbf{c}_j||}\right) \leq \Delta \alpha \end{cases} \quad (5)$$

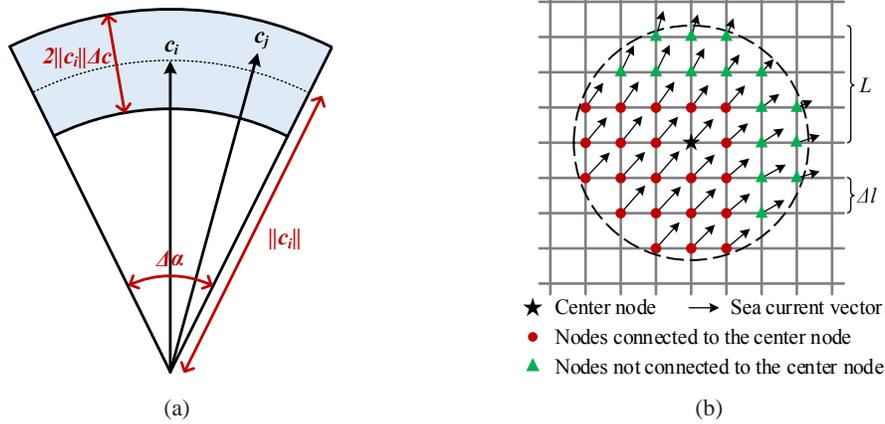


Fig. 3. Generating a directed graph by using the concept of similarity of sea current vectors. (a) Definition of similarity of sea current vectors. (b) Example of how one node (the center node) connects to the neighbors via (5), where $L = 3.3\Delta l$, $\Delta c = 0.1$ and $\Delta\alpha = 2^\circ$. Note that the outdegree of the center node is variable, depending on parameters L , Δc , and $\Delta\alpha$.

where “ \cdot ” and “ $\| \cdot \|$ ” denote dot product and Euclidean norm, respectively. For zero case of either c_i or c_j , we still assume they are similar.

In addition, for each node, suppose there is a circle with a radius of L around that center node, representing the maximum connection length. Combined with (5), the center node is connected to all nodes within the circle that have a similar sea current vector to that of the center node, as in the example shown in Fig. 3b. As a result, each node in the generated directed graph has a variable number of connected neighbors, which not only increases the possible moving directions, but also ensures the consistency of sea current during the motion.

B. Reachability

As mentioned above, the edge between two nodes in the directed graph shows rational spatial relationship and sea current consistency. Nevertheless, it does not guarantee the reachability from one node to the other due to (1) and (2). Suppose at the t th wavefront search, there is a search-front node n_j connecting to its neighbor n_k . Given the sea current vector ($c(n_j)$ and $\alpha(n_j)$), the moving direction Φ_0 from n_j to n_k , and n_j 's predecessor n_i which results in the minimal travel time $t^*(n_j)$ from the initial node to n_j , **reachability** (from n_j to n_k) is to check whether:

- (i) there is a proper thrust orientation angle $\theta(n_j)$ at n_j such that the resultant velocity vector ($u(n_j)$ and $\phi(n_j)$) is exactly towards the neighbor n_k , i.e., $\phi(n_j) = \Phi_0$ and $u(n_j) > 0$;

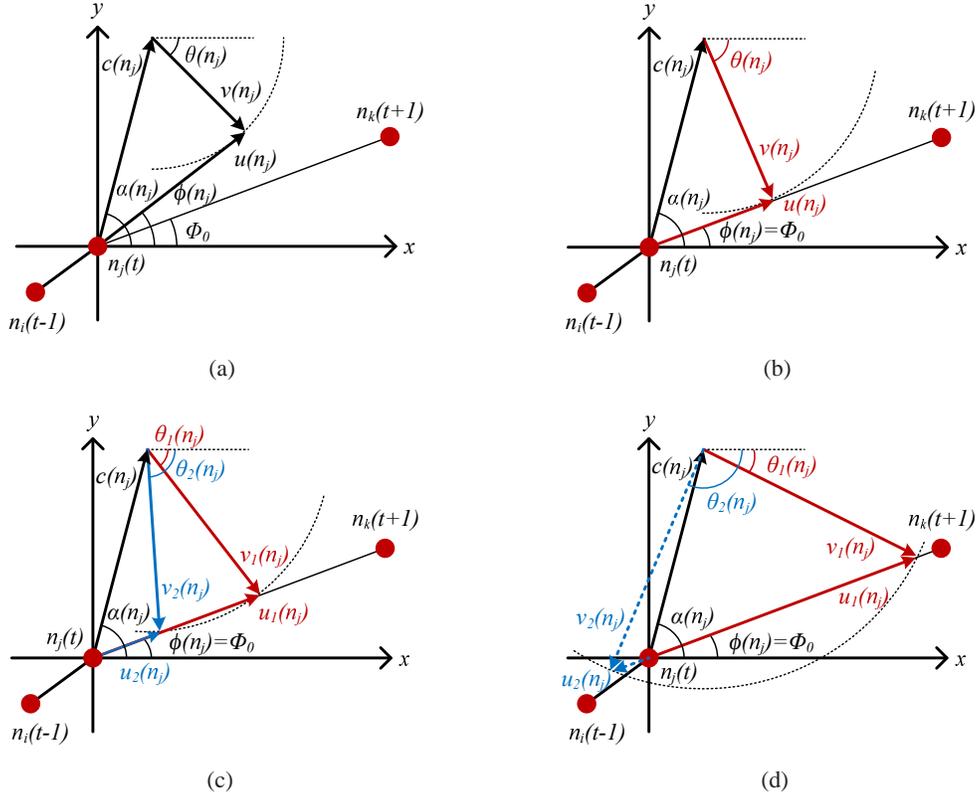


Fig. 4. Geometric illustration of finding $\theta(n_j)$ at n_j that leads to resultant moving direction $u(n_j)$ towards n_k . Taking the end position of the vector $(c(n_j), \alpha(n_j))$ as the center of a circle and the magnitude of v as the radius, there are four cases of solutions, depending on the intersection(s) with the straight line between n_j and n_k . (a) No solution. (b) Unique solution. (c) Two valid solutions. (d) Two solutions with one valid and the other invalid (represented as the dashed line).

- (ii) the thrust orientation changing from $\theta(n_i)$ at n_i to $\theta(n_j)$ at n_j must satisfy (2). Here, we simplify the process such that the change of thrust orientation happens only when the vehicle arrives at a new node, and the process is untimed.

From a geometric perspective, the solution to (i) can be divided into four cases, as illustrated in Fig. 4. Indeed, substituting (3) into (1), multiplying the two sides of the equations and rearranging the result yield the reachability constraint for (i):

$$\sin(\phi(n_j) - \theta(n_j)) = \frac{c(n_j)}{v} \sin(\alpha(n_j) - \phi(n_j)) \quad (6)$$

The solution relies on the right side value of (6). When its absolute value is bigger than 1, (6) has no solution (see illustration in Fig. 4a). If its absolute value equals 1, the unique solution as depicted in Fig. 4b is expressed as:

$$\theta(n_j) = \begin{cases} \phi(n_j) - \pi/2 & \text{if } \alpha(n_j) > \phi(n_j) \\ \phi(n_j) + \pi/2 & \text{otherwise} \end{cases} \quad (7)$$

Otherwise, there are two solutions for (6) (see illustration in Fig. 4c):

$$\theta(n_j) = \begin{cases} \phi(n_j) - \arcsin\left(\frac{c(n_j)}{v} \sin(\alpha(n_j) - \phi(n_j))\right) \\ \phi(n_j) + \arcsin\left(\frac{c(n_j)}{v} \sin(\alpha(n_j) - \phi(n_j))\right) - \pi \end{cases} \quad (8)$$

Note from Fig. 4d that for the case of two solutions, one of them may be invalid due to $u(n_j) \leq 0$, resulting in an opposite moving direction away from n_k . Therefore, it is necessary to bring (8) back into (1) and (3) to verify the result, and to calculate $u(n_j)$ as well.

If (7) or (8) exists, the travel time from n_j to n_k will be:

$$t(n_j, n_k) = \frac{\text{dist}(n_j, n_k)}{u(n_j)} \quad (9)$$

where $\text{dist}(\cdot)$ represents the connection length between the two nodes. By considering the simplification in (ii), thrust orientation constraint is expressed as:

$$|\theta(n_j) - \theta(n_i)| \leq \delta t(n_j, n_k) \quad (10)$$

which indicates only when thrust orientation difference is smaller than $\delta t(n_j, n_k)$, moving from n_j to n_k is considered reachable. Once the reachability to n_k is determined, n_k 's arrival time $t(n_k) = t^*(n_j) + t(n_j, n_k)$ is obtained, which can be used for collision risk assessment; see Section III-C.

Remark 2: From (9) and (10), longer connection length dist leads to the constraint in (10) being more relaxed. It is thus possible that for the two neighbors of a single node with the same Φ_0 but different dist , the closer neighbor is unreachable but the distant neighbor is reachable. This phenomenon follows the fact of steering a vehicle with an upper bound on curvature [40].

C. Collision Avoidance

Collision avoidance is associated with both static obstacles such as oil platforms and dynamic obstacles like TSs. Following the notations used in Section II-B, Table I lists the parameters of the two types of obstacles used throughout this section. Note that static obstacles are simplified as a circle, while dynamic obstacles are represented as a rectangle.

TABLE I
NOTATION OF OBSTACLES' PARAMETERS

Obstacle type	Parameter	Description
Static obstacles	X_s	Obstacles' 2D position
	D_s	Obstacles' dimensions (in radius)
	\mathcal{S}	The set of the whole static obstacles
Dynamic obstacles (TSs)	X_d	Obstacles' 2D position
	Φ_d	Obstacles' orientation
	V_d	Obstacles' velocity magnitude over ground
	D_d	Obstacles' dimension (length \times width)
	\mathcal{D}	The set of the whole dynamic obstacles

Collision risk is assessed on n_k once its reachability from n_j has been determined. This indicates its arrival time $t(n_k)$ from initial state and transition state $P(t(n_k)) = (X_{n_k}, u(n_k), \theta(n_j), \phi(n_j))$ are known (X_{n_k} is the 2D position of n_k). Given the OS's 2D dimension D_0 (length \times width) and the sets $\mathcal{S} := (X_s, D_s)$ and $\mathcal{D} := (X_d, \Phi_d, V_d, D_d)$, collision risk assessment on n_k is expressed in an accumulative way:

$$\begin{aligned}
 R(n_k) &= Col(p(t(n_k)), D_0, \mathcal{S}) + Col(p(t(n_k)), D_0, \mathcal{D}) \\
 &= \sum_{s_i \in \mathcal{S}} col(s_i) + \sum_{d_i \in \mathcal{D}} col(d_i)
 \end{aligned} \tag{11}$$

where $Col(\cdot)$ and $col(\cdot)$ are the functions of collision risk assessment for one type of obstacles and individual obstacle, respectively.

For conciseness, the shapes of the OS, the TSs, and the static obstacles are converted from size format D_0 , D_s , and D_d into vertex format Q_0 , Q_s , and Q_d , respectively. Let C_s and C_d be the arrays of clearance distances for the two types of obstacles, wherein $\forall s_i \in \mathcal{S}$, $C_s(s_i) > D_s(s_i)$ and $\forall d_i \in \mathcal{D}$, $C_d(d_i) > \|D_d(d_i)\|$. They are the parameters that represent virtual circles around each of the obstacles. Only when the OS moves into any of these circles, is collision risk considered.

For every $s_i \in \mathcal{S}$, let $dist(s_i) = \|X_{n_k} - X_s(s_i)\|$ for concise purpose. The collision risk between s_i and the OS is defined as a piecewise function:

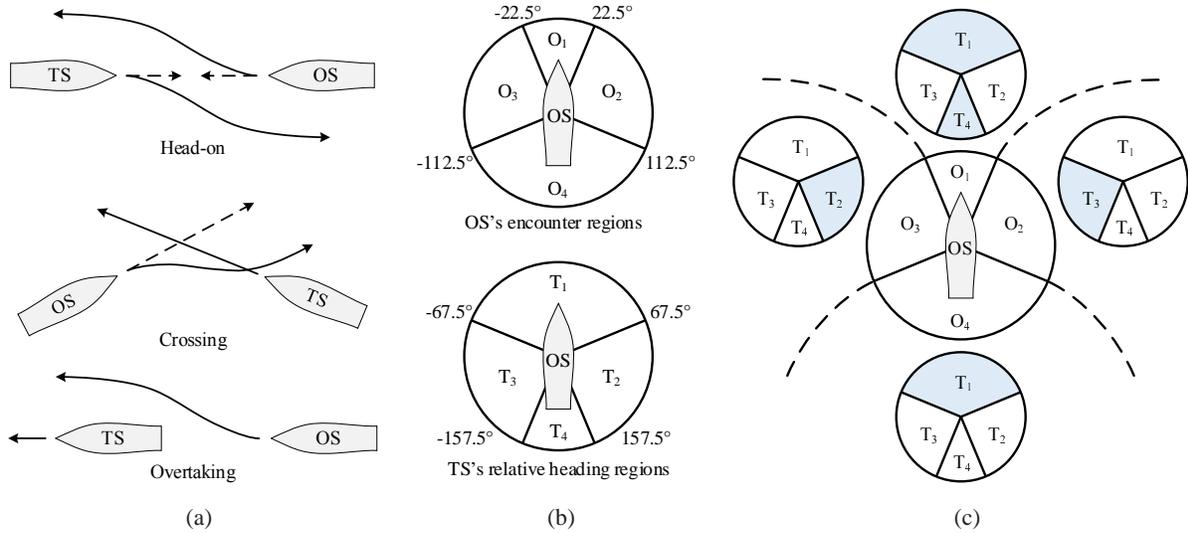


Fig. 5. Encounter situation for dynamic obstacles. (a) Encounter types in COLREGs. (b) Classification of encounter regions. (c) High risk of collision on encounter situations O_1T_1 , O_1T_4 , O_2T_3 , O_3T_2 , and O_4T_1 .

$$col(s_i) = \begin{cases} \infty & \text{if } poly(X_{n_k} + R_0 Q_0, Q_s(s_i)) < 0 \\ e^{\frac{C_s(s_i) - D_s(s_i)}{dist(s_i) - D_s(s_i)}} & \text{if } D_s(s_i) < dist(s_i) < C_s(s_i) \\ 1 & \text{if } dist(s_i) \geq C_s(s_i) \end{cases} \quad (12)$$

wherein R_0 is the rotation matrix rotating points Q_0 through an angle of $\phi(n_j)$ about the origin of the OS; $poly(\cdot)$ is the function to detect overlap of convex polygons, with values smaller than zero indicating polygons are overlapping [42]. The three cases in (12) stand for the risk factor between s_i and the OS at n_k where they collide, near far and far away, respectively.

For dynamic obstacles (TSs), predicting their future motion is needed. One of the simplest predictions is straight line trajectories [37]. When the OS is at n_k , for every $d_i \in \mathcal{D}$, its position is updated from the initial state:

$$X_d(d_i, t) = X_d(d_i, 0) + V_d(d_i) t \begin{bmatrix} \cos(\Phi_d(d_i)) \\ \sin(\Phi_d(d_i)) \end{bmatrix} \quad (13)$$

Accordingly, based on the states of both the OS and d_i at time $t(n_k)$, their encounter type can be determined. In COLREGs, the encounter situations between the OS and the TS are divided into three types: head-on, crossing, and overtaking, as illustrated in Fig. 5a. The vehicles act either to “give-way” or “stand-on”. The difference is the “give-way” vehicle should alter its course,

TABLE II
ENCOUNTER TYPES AND OS ACTIONS UNDER ENCOUNTER SITUATIONS

Encounter type	Encounter situation	OS action
Head-on	O_1T_4	Give way
	O_2T_4, O_3T_4	Stand on
Crossing	O_2T_3	Give way
	$O_1T_2, O_1T_3, O_2T_2, O_3T_2$	Stand on
	$O_3T_3, O_4T_2, O_4T_3, O_4T_4$	
Overtaking	O_1T_1	Give way
	O_2T_1, O_3T_1, O_4T_1	Stand on

whereas the “stand-on” vehicle is suggested to maintain its course and speed. Inspired by [41] and according to rules 13-18 in COLREGs, we categorize the position of d_i with respect to the position and heading of the OS into regions from O_1 to O_4 , and the relative heading of d_i by comparing $\Phi_d(d_i)$ and $\phi(n_j)$ in regions T_1 to T_4 , as shown in Fig. 5b. In total, there are 16 encounter situations (ESs) represented in O_iT_j format. Table II lists these ESs and corresponding action by the OS. Five of them, as illustrated in Fig. 5c, are considered to be at high risk of collision in this paper. Note that even though the OS is the “stand-on” vehicle in situations O_3T_2 and O_4T_1 , the probability of collision still exists as the TS may fail to take action.

Collision risk assessment for TSs contains two phases. The first phase checks the current collision—when the OS arrives at n_k :

$$r_1 = \begin{cases} \infty & \text{if } \text{poly}(X_{n_k} + R_0Q_0, X_d(d_i) + R_dQ_d(d_i)) < 0 \\ 1 & \text{otherwise} \end{cases} \quad (14)$$

where R_d stands for the rotation of $Q_d(d_i)$ through an angle of $\Phi_d(d_i)$ about the origin of d_i . The second phase is a variant of the closest point of approach, which considers the near-future collision by assuming the OS continues to move along the direction $\phi(n_j)$ with a forward speed $u(n_k)$, and evaluates the risk in terms of the distance $\text{dist}(d_i, t)$ between the OS and d_i :

$$\begin{cases} X(t) = X_{n_k} + u(n_k) t \begin{bmatrix} \cos(\phi(n_j)) \\ \sin(\phi(n_j)) \end{bmatrix} \\ \text{dist}(d_i, t) = \|X(t) - X_d(d_i, t + t(n_k))\| \end{cases} \quad (15)$$

where $t \geq 0$ denotes the future time. By combining (13) and (15), $dist(d_i, t)$ becomes a convex function with respect to t . Its minimum $\min(dist(d_i, t))$ exists when either $t = 0$ or $t > 0$, indicating that the OS and d_i will either depart forever, or come closer and then depart, respectively. The latter case presents more danger, especially for the encounter situations in Fig. 5. Therefore, the second phase risk assessment is defined as follows:

$$r_2 = \begin{cases} 1 & \text{if } dist(d_i, 0) > C_d(d_i) \\ \gamma e^{\frac{C_d(d_i)}{\min dist(d_i, t) + \epsilon}} & \text{otherwise} \end{cases} \quad (16)$$

where ϵ is a small positive constant; and γ is a piecewise function:

$$\gamma = \begin{cases} 1 & \text{if } \operatorname{argmin}_t dist(d_i, t) = 0 \\ p & \text{if } \operatorname{argmin}_t dist(d_i, t) \neq 0 \text{ \& ES in Fig. 5c} \\ q & \text{otherwise} \end{cases} \quad (17)$$

where p and q are constants satisfying $p \gg q > 1$. Consequently, collision risk for d_i is expressed as:

$$col(d_i) = \max(r_1, r_2) \quad (18)$$

The total collision risk in (11) will be used as a kind of cost for the search among the wavefront nodes, see Section III-D.

D. Cost Evaluation

The constraints in (4) are converted into costs for evaluation during the wavefront search. Besides costs from Section III-B and III-C, the costs U_0 and U_{t_f} for user-defined initial and final states (denoted as $P_0 := (X_0, \theta_0, \phi_0)$ and $P_{t_f} := (X_{t_f}, \theta_{t_f}, \phi_{t_f})$, respectively) are defined as a function taking effect only on a specific temporal-spatial domain:

$$U_0 = \begin{cases} e^{\frac{\|z - z_0\|}{t + \epsilon}} & \text{if } 0 < t < T_0 \\ 1 & \text{otherwise} \end{cases} \quad (19)$$

and

$$U_{t_f} = \begin{cases} e^{\frac{\|z - z_{t_f}\|}{\|X - X_{t_f}\| + \epsilon}} & \text{if } \|X - X_{t_f}\| < C_{t_f} \\ 1 & \text{otherwise} \end{cases} \quad (20)$$

Algorithm 1: Dijkstra-like solver for close-range maneuvering problem

Data: $P_0 = (X_0, z_0)$ and $P_{t_f} = (X_{t_f}, z_{t_f})$

Result: Nodes on the optimal path from X_0 to X_{t_f}

\mathcal{N} := set of nodes;

mapping X_0 to $n_0 \in \mathcal{N}$, and X_{t_f} to $n_{t_f} \in \mathcal{N}$;

$N_C(n_x, n_y) := n_x$ connecting to n_y by (5);

$\mathcal{N}_B(n_x) := \{n_y \mid N_C(n_x, n_y)\}$;

$C(n_x) :=$ cost in path from n_0 to n_x ;

$prev(n_x) :=$ previous node of n_x in path from n_0 ;

$\mathcal{K} := \{n_x \in \mathcal{N} \mid cost(n_x) \text{ is known}\}$;

$\mathcal{U} := \{n_x \in \mathcal{N} \mid cost(n_x) \text{ is unknown}\}$;

$cost(n_0) \leftarrow 0$;

for $n_i \in \mathcal{N} - \{n_0\}$ **do**

| $cost(n_i) \leftarrow \infty$;

$\mathcal{K} \leftarrow \emptyset$ and $\mathcal{U} \leftarrow \mathcal{N}$;

while $\mathcal{U} \neq \emptyset$ **do**

| $n_j \leftarrow \operatorname{argmin}_{n_j \in \mathcal{U}} C(n_j)$;

| $\mathcal{K} \leftarrow \mathcal{K} \cup \{n_j\}$ and $\mathcal{U} \leftarrow \mathcal{U} - \{n_j\}$;

| **for** $n_k \in \mathcal{N}_B(n_j)$ **do**

| | **if** $reachable(n_j, n_k)$ by (6) and (10) **then**

| | | $cost(n_j, n_k) = t(n_j, n_k) \cdot R(n_k) \cdot U_0 \cdot U_{t_f}$ by (9), (11), (19) – (21);

| | **if** $C(n_k) > C(n_j) + cost(n_j, n_k)$ **then**

| | | $C(n_k) \leftarrow C(n_j) + cost(n_j, n_k)$;

| | | $prev(n_k) = n_j$;

return $prev$;

, or

$$U_{t_f} = \begin{cases} e^{-\frac{\|X-X_{t_f}\|}{C_{t_f}}} & \text{if } \|X-X_{t_f}\| < C_{t_f} \\ 1 & \text{otherwise} \end{cases} \quad (21)$$

for the case of no orientation constraints on the final state, where T_0 represents a fixed time period; C_{t_f} stands for the radius of the circle centered at X_{t_f} ; $z \in \{\theta, \phi\}$ is the thrust/ship orientation, corresponding to the user-defined parameters of $z_0 \in \{\theta_0, \phi_0\}$ and $z_{t_f} \in \{\theta_{t_f}, \phi_{t_f}\}$.

Once the constraints in (4) are quantified as costs, an approximate solution to (4) can be realized through a wavefront search, as depicted in Algorithm 1. Accordingly, the path with the lowest cost (if it exists) is expressed as a set of nodes with a fixed sequence obtained by reverse iteration of the returned indexing vector *prev*.

IV. PATH SMOOTHING

This section introduces a new smoothing method in conjunction with parameter optimization technology for an exploration of a near-optimal path around the discrete solution obtained from Section III.

A. Bézier Curve Connection

Bézier curves are a type of parametric curve that has been widely used in computer graphics [43]. The parametric Bézier curve of degree M is usually expressed as:

$$G(\eta) = \sum_{i=0}^M b_{i,M}(\eta) B_i, \quad \eta \in [0, 1] \quad (22)$$

where η is the parameter describing the interpolation of the Bézier curve; B_i denotes the i th control point of G ; and $b_{i,M}(\eta)$ represents the i th Bernstein polynomial given by:

$$b_{i,M}(\eta) = \binom{M}{i} \eta^i (1-\eta)^{M-i} \quad (23)$$

A Bézier curve is characterized by: (1) it starts at B_0 and ends at B_M ; and (2) $\overline{B_0 B_1}$ and $\overline{B_{M-1} B_M}$ are the tangent directions of G at B_0 and B_M , respectively.

Given there are H points from O_1 to O_H in the discrete path obtained by Algorithm 1, path smoothing involves constructing multiple Bézier curves around the discrete path and joining them smoothly, as shown in Fig. 6. Our previous work introduced a method to join cubic Bézier curves ($M=3$) with C^1 continuity [16]. Nevertheless, considering the constraint (2) and its relation to

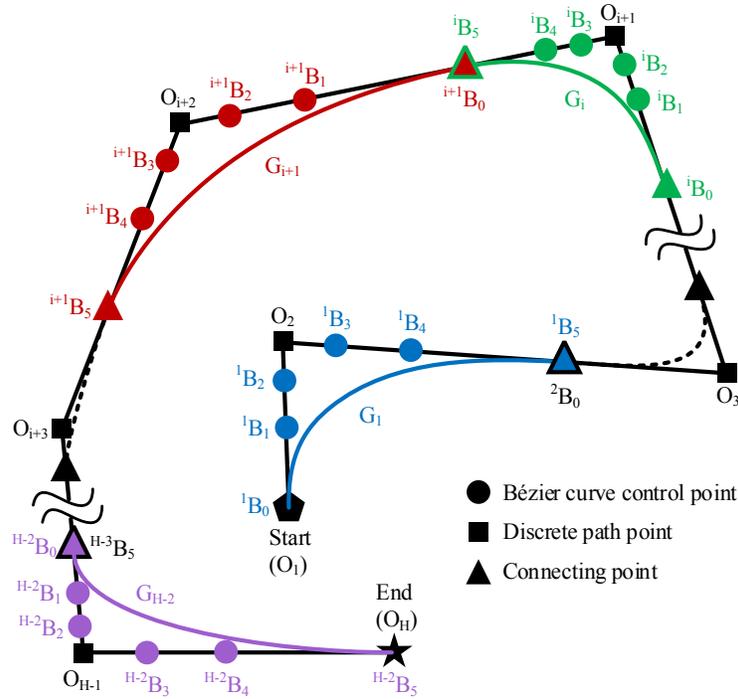


Fig. 6. Connection of 5th-order Bézier curves with C^2 continuity. Each color line corresponds to a Bézier curve.

curvature, connecting Bézier curves with C^2 continuity is needed.¹ The following introduces a method to join 5th-order Bézier curves ($M=5$) with C^2 continuity.

There is a type of control points called connecting points that serve as the end of one Bézier curve and the beginning of the other Bézier curve, as illustrated in Fig. 6. Suppose the connecting points are on each segment of the discrete path, excluding the segments with O_1 and O_H involved. Thus in total, there are $H-3$ connecting points used for joining $H-2$ Bézier curves on the discrete path. For each Bézier curve G_i , apart from the control points at each end (iB_0 and iB_5 , respectively), the other four control points are designed to be evenly divided into two groups and to be limited into the two adjacent segments of the discrete path, that is, iB_1 and iB_2 on $\overline{{}^iB_0O_{i+1}}$, while iB_3 and iB_4 on $\overline{O_{i+1}{}^iB_5}$. The purpose is to decrease geometric constraints to connect two Bézier curves (see Remarks 3 and 4).

From (22), the 1st- and 2nd-order derivatives of G_i with respect to η are determined by its control points [45]:

¹As one of the criteria for path smoothness evaluation, the parametric C^1 continuity indicates the derivative of $G(\eta)$ in (22) w.r.t. η is continuous; C^2 continuity is stricter, which requires both the first- and second-order derivatives of $G(\eta)$ are continuous [44].

$$\dot{G}_i(\eta) = M \sum_{i=0}^{M-1} b_{i,M-1}(\eta) (B_{i+1} - B_i) \quad (24)$$

$$\ddot{G}_i(\eta) = M(M-1) \sum_{i=0}^{M-2} b_{i,M-2}(\eta) (B_{i+2} - 2B_{i+1} + B_i) \quad (25)$$

Since the polynomial function $b_{i,M}(\eta)$ is continuous, both $\dot{G}_i(\eta)$ and $\ddot{G}_i(\eta)$ are continuous. Hence, G_i has C^2 continuity for all $\eta \in [0, 1]$.

In order to join the $(H-2)$ Bézier curves with C^2 continuity, for any two adjacent G_i and G_{i+1} , their 1st- and 2nd-order derivatives at the connecting point iB_5 (${}^{i+1}B_0$) must be continuous. That is:

$$\begin{cases} {}^iB_5 = {}^{i+1}B_0 \\ \dot{G}_i(1) = \dot{G}_{i+1}(0) \\ \ddot{G}_i(1) = \ddot{G}_{i+1}(0) \end{cases} \quad (26)$$

Taking (24) and (25) into (26) with $M=5$ yields the geometric requirements to achieve C^2 continuity:

$$\begin{cases} {}^{i+1}B_1 = 2{}^iB_5 - {}^iB_4 \\ {}^{i+1}B_2 = 4{}^iB_5 - 4{}^iB_4 + {}^iB_3 \end{cases} \quad (27)$$

Remark 3: Equation (26) reveals the dependence of G_{i+1} on G_i . Note because the control points iB_3 , iB_4 and iB_5 are set on the segment $\overline{O_{i+1}O_{i+2}}$, ${}^{i+1}B_1$ and ${}^{i+1}B_2$ represented as the linear combination of the three control points should be also located on the same segment, or on its extension.

Remark 4: Locating low-order Bézier curves' control points ($M < 5$) on the discrete path cannot guarantee C^2 continuity. On the one hand, there are M control points on each segment of the discrete path due to the end-to-end connection between Bézier curves (see Fig. 6 for example). On the other hand, similar constraints to (27) can be obtained in which five control points are involved to achieve C^2 continuity. This implies at least one involved control point is on neither the segment nor its extension. Therefore, the constraint cannot be satisfied through linear combination.

B. Path Parameterization and Optimization

In the construction of C^2 continuous Bézier curves, there are $3H-5$ points determining the smoothness of the path, including $H-3$ connecting points, and $2(H-1)$ control points (two control points on each segment of the discrete path, see Fig. 6). Here we use percentage representation instead of positions of these points to parameterize the path. For connecting points, let $\tau_i \in (0, 1)$ be the percentage of the line segment. Thus, the position of the connecting point can be expressed as:

$${}^iB_5 = O_{i+1}\tau_i + O_{i+2}(1 - \tau_i), \quad i = 1, 2 \dots H-3 \quad (28)$$

In addition, parameters $\lambda_{1i} \in (0, 1)$ and $\lambda_{2i} \in (0, 1)$ are utilized to represent the two control points on each segment:

$$\begin{cases} {}^iB_3 = O_{i+1}\lambda_{1i} + {}^iB_5(1 - \lambda_{1i}) \\ {}^iB_4 = {}^iB_3\lambda_{2i} + {}^iB_5(1 - \lambda_{2i}) \end{cases} \quad i = 1, 2 \dots H-3 \quad (29)$$

Similar representations are applied to the two control points on the segment at each end (1B_1 , 1B_2 , ${}^{H-2}B_3$ and ${}^{H-2}B_4$).

Once the smooth path is parameterized by the $3H-5$ parameters, which are $[\tau, \lambda_1, \lambda_2]$, GA is used to evolve these parameters for the optimal smooth path that has the lowest risk of collision. Note that the user-defined initial and final states will not be changed due to the properties of Bézier curves. Therefore, the costs U_0 and U_{tf} in (19) – (21) are ignored in the optimization process. For each individual in the GA generation, the corresponding arrival time of all the points on the smooth path can be estimated. Because the arrival time of the discrete path is known from Section III-B, it is intuitive to estimate the arrival time of the connecting point in Fig. 6 via the geometric relationship between the adjacent discrete points:

$$t({}^iB_5) = t(O_{i+1}) + (t(O_{i+2}) - t(O_{i+1})) \frac{\|{}^iB_5 - O_{i+2}\|}{\|O_{i+1} - O_{i+2}\|} \quad (30)$$

The rest points on the smooth path can be calculated in the same way, but use the curve length rather than segment length to estimate the arrival time. The corresponding vehicle speed can thus be obtained if we assume the speed along the curve of two adjacent connecting points is constant.

The curvature of the discrete path can be calculated in the first stage of the scheme by using its definition $\rho_1=|d\phi/ds|$ (here s denotes the curve length). Also, it is straightforward to obtain curvature for the i th Bézier curve by using (24) and (25) for comparison:

$$\rho_2 = \frac{|\dot{G}_x \ddot{G}_y - \dot{G}_y \ddot{G}_x|}{\|\dot{G}\|^3} \quad (31)$$

where $\dot{G}=(\dot{G}_x, \dot{G}_y)$ and $\ddot{G}=(\ddot{G}_x, \ddot{G}_y)$. In GA, if the curvatures of the smoothed paths are smaller than those of their counterparts ρ_1 , the corresponding individuals with low cost from collision avoidance in (11) are selected to perform GA operation, including crossover and mutation to breed a new generation. The process will be repeated until the termination condition, such as the maximum number of generations, has been reached. After the optimization, the resulting path is close to the discrete solution but smooth with C^2 continuity.

V. EXPERIMENTS

To verify the correctness and effectiveness of the path generated by the proposed approach, a benchmark test and a series of experiments, including initial and final state constraint, thrust constraint and collision avoidance, were conducted in a computer equipped with 2.60 GHz i7-6600U CPU and 8 GB RAM.

A. Comparison with Benchmark

The Zermelo's ship steering problem is a well-known optimization problem that can be solved analytically [46]. Here we use its solution as the benchmark and compare it with the result from our approach.

Ship dimension is negligible in this problem. Suppose a ship is initially positioned at $X_0=(3.66, -1.86)$ with a thrust orientation $\theta_0=105^\circ$. Zermelo's problem is to steer the ship at a constant speed $v=1\text{ m/s}$ through a sea current field $\mathbf{c}=(-y, 0)$ to reach the position $X_{t_f}=(0, 0)$ in minimum travel time. In other words, the problem can be expressed as a simplification of (4):

$$\begin{aligned} & \min_{\theta} t_f \\ & s.t. \begin{cases} Eq. (1) \\ P(0) = (X_0, \theta_0) \\ P(t_f) = X_{t_f} \\ t \in [0, t_f] \end{cases} \end{aligned} \quad (32)$$

TABLE III
PARAMETERS OF WAVEFRONT SEARCH IN ZERMELO'S PROBLEM

Parameter	Value	Description
$m \times n$	401×201	Number of nodes in the directed graph
Δl	$0.02m$	Nodes' intervals
L	$0.2m$	Nodes' maximum connection length
Δc	0.1	Allowed current magnitude change in (5)
$\Delta \alpha$	2°	Allowed current orientation change in (5)
T_0	$0.5s$	Time threshold in (19)
C_{t_f}	$0.3m$	Distance threshold in (20)

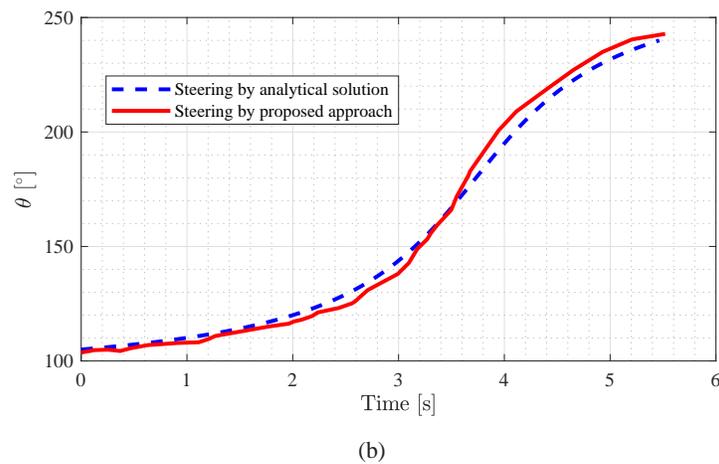
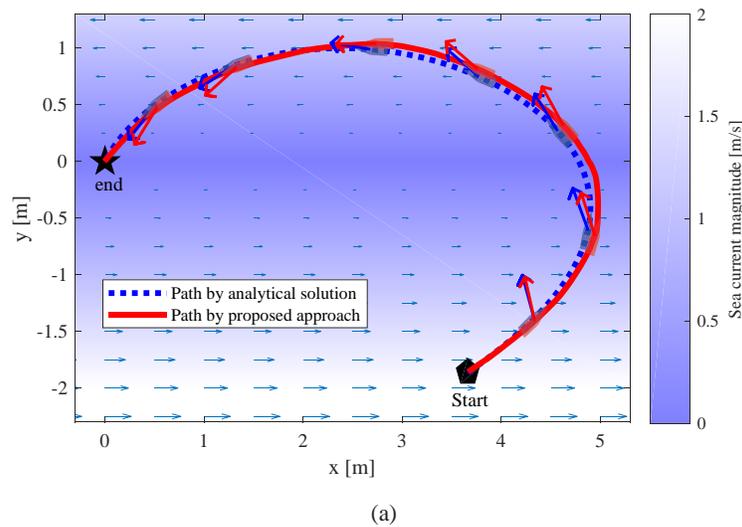


Fig. 7. Comparison between the analytical solution and the proposed approach in Zermelo's problem. (a) Comparison of the planning paths. The arrows represent thrust orientations. (b) Comparison of thrust orientation.

The two-stage scheme was carried out and the parameter settings for the wavefront search are listed in Table III. Since there are neither thrust constraints like (2) nor obstacles in the application, Algorithm 1 with simplified cost from (19) and (21) was conducted, resulting in a discrete path containing $H=40$ points. After 100 generations of evolution through GA, the path was further optimized toward acceptable smoothness. It took 156s to complete the experiment.

From Fig. 7a, the shape of the computed path is similar to that of the analytical path. The vehicle is drifted more to the right at the beginning due to the accumulated error of steering caused by the granularity of the directed graph. According to Fig. 7b, however, the degree of steering is compensated back after about 3.5 s when the vehicle passes (2.7, 1.0) using a thrust angle of 180° . Furthermore, the travel time along the computed path is 5.52 s. Compared to 5.46 s, the minimum travel time in theory, the time error is as low as 1.09%.

The similarity in terms of both path shape and travel time demonstrates that the two-stage trajectory planning scheme is a suitable approach for generating a near time-optimal path.

B. Path Planning under Initial and Final State Constraints

Gyre flow is one type of spatially complex sea current caused by wind movements. In this section, the proposed approach was applied in simulated gyres given by Cartesian format:

$$\begin{cases} c_x(x, y) = -\sin(x\pi/250) \cos(y\pi/250) \\ c_y(x, y) = \cos(x\pi/250) \sin(y\pi/250) \end{cases} \quad (33)$$

The working space is set to $500\text{ m} \times 500\text{ m}$. There are 201×201 nodes evenly distributed in this scope. A directed graph was generated by setting $L=10\text{ m}$, $\Delta c=0.1$ and $\Delta\alpha=2^\circ$, respectively. Suppose the OS has a dimension of $D_0=82\text{ m} \times 23\text{ m}$, and its initial and final positions are both at the center of a gyre: $X_0=(125\text{ m}, 125\text{ m})$ and $X_{t_f}=(375\text{ m}, 375\text{ m})$.

The optimization problem here is the same as (32), but differs from i) adding (2) with thrust angular speed limit $\delta=18^\circ/\text{s}$; and ii) setting a stricter initial and final states constraint: $P(0)=(X_0, \phi_0)$ and $P(t_f)=(X_{t_f}, \phi_{t_f})$. Note that the ship heading is the main concern in this experiment. The related parameters for initial and final states are set as $T_0=10\text{ s}$ and $C_{t_f}=20\text{ m}$. Traversals of initial and final states within $[-180^\circ, 180^\circ] \times [-180^\circ, 180^\circ]$ under different vehicle velocity $v \in (0, 1]\text{ m/s}$ were conducted.

For conciseness, Fig. 8 shows only optimal trajectories under three different vehicle velocities, as well as the corresponding results for the most time efficient path under that velocity. Each

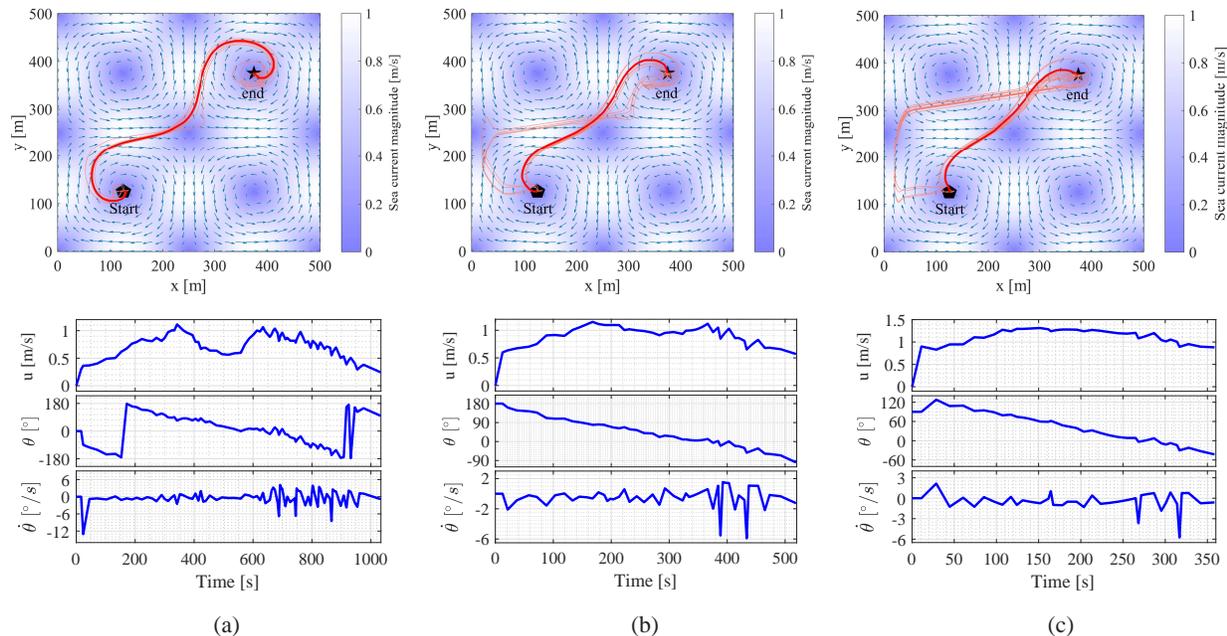


Fig. 8. Solutions for the “initial and final state constraints” optimization problem with respect to different vehicle velocities. (a) $v=0.3\text{ m/s}$. (b) $v=0.6\text{ m/s}$. (c) $v=0.9\text{ m/s}$.

TABLE IV
STATISTICS OF MINIMUM-TIME PATHS FOR THE “INITIAL AND FINAL STATE CONSTRAINTS” OPTIMIZATION PROBLEM

v [m/s]	ϕ_0 [°]	ϕ_{t_f} [°]	t_f [s]	s [m]	\bar{u} [m/s]
0.2	0	120	1797.78	1123.68	0.62
0.3	0	135	1033.29	708.80	0.69
0.4	180	180	775.23	580.67	0.75
0.5	180	-100	615.78	495.76	0.81
0.6	180	-80	518.92	462.13	0.89
0.7	180	-60	449.71	443.41	0.97
0.8	180	-50	398.66	431.33	1.08
0.9	90	-30	358.42	399.28	1.11
1.0	90	-30	324.02	396.27	1.22

successful search took about 120s of computation time. It is worth noting that among the optimal trajectories, $\phi_0 \in \{0^\circ, 90^\circ, 180^\circ\}$ dominates the majority; and the applicable range of ϕ_{t_f} increases with the growth of vehicle velocity. Table IV lists the results of minimum-time paths among the solutions of all possible initial and final state constraints for each specific $v \in (0, 1] \text{ m/s}$. We found that when $v < 0.2 \text{ m/s}$, the target position is unattainable. But for $v \geq 0.2 \text{ m/s}$, the vehicle

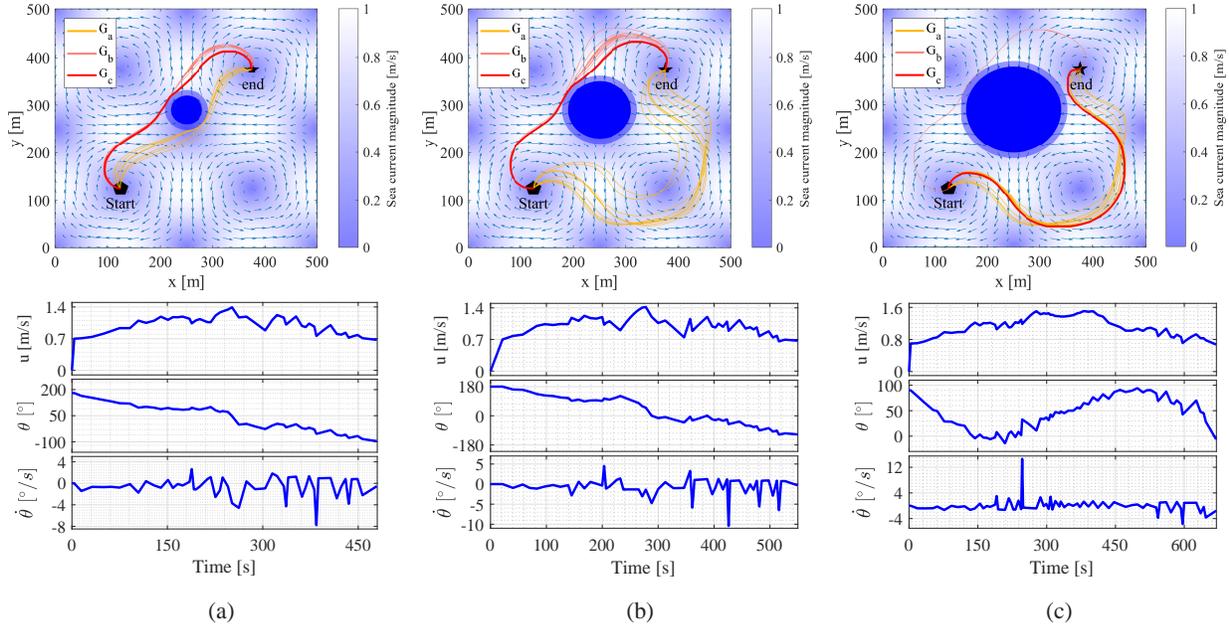


Fig. 9. Solutions for the “thrust angular speed limit” optimization problem with respect to different dimensions of static obstacles. (a) $D_s(s_1)=30\text{ m}$. (b) $D_s(s_1)=60\text{ m}$. (c) $D_s(s_1)=90\text{ m}$.

can ride the sea current (see the averaged total speed \bar{u} in Table IV) to reach the destination while satisfying (2).

Note that not all combinations of ϕ_0 and ϕ_{t_f} have a solution. As described in Remark 1, this is because too-strict constraints make the problem unsolvable. In fact, setting the constraint on the final state is more practical, since it involves subsequent maritime operations after the vehicle has arrived at the destination; in contrast, adjusting the initial state becomes simpler, as the working space near the ship is wide and open.

C. Path Planning under Thrust Angular Velocity Limit

This section investigates the influence of thrust angular velocity limit δ for path planning in gyre flow. The scene in Section V-B is adopted (including X_0 , X_{t_f} and D_0 of the OS, and gyre parameters). A static obstacle s_1 with a variable radius $D_s(s_1)$ located at $X_s(s_1)=(250\text{ m}, 290\text{ m})$ is added to block the minimum-time paths in Fig. 8 to the destination. The clearance distance for s_1 is set to half the width of the OS wider than $D_s(s_1)$, i.e., $C_s(s_1)=D_s(s_1)+11.5\text{ m}$, so as to prevent collision when the OS passes by s_1 . As there are no heading constraints on both initial and final states, the optimization problem with thrust angular velocity limit is expressed

as:

$$\begin{aligned} & \min_{\theta} t_f \\ & s.t. \begin{cases} Eq. (1) (2) \\ 0 < Col(P(t), D_0, s_1) < \infty \\ P(0) = X_0 \\ P(t_f) = X_{t_f} \\ t \in [0, t_f] \end{cases} \end{aligned} \quad (34)$$

The experiment was conducted under various obstacle's dimensions $D_s(s_1) \in [30 m, 90 m]$ by setting vehicle parameters $v=0.7 m/s$ and $\delta \in (0^\circ/s, 18^\circ/s]$. The computation time for successful search depends on $D_s(s_1)$ and δ , ranging from 101s to 149s. Fig. 9 illustrates the time optimal paths for three cases of $D_s(s_1)$. With the growth of δ , these paths in each case can be roughly divided into three categories:

- G_a : When δ is small, which indicates that the OS has a poor steering ability, the OS is able to pass by s_1 from below to reach X_{t_f} in longer travel time.
- G_b : Improving the OS's steering ability by applying a higher value of δ to the planner results in better solutions. The OS travels to X_{t_f} in shorter time if it can pass s_1 from above.
- G_c : Further increasing δ cannot help to find trajectories with shorter travel time. This means the angular speed limit no longer has any effect. Therefore, G_c is the set of identical solutions representing the minimum-time path.

The lower part of Fig. 9 depicts the maneuvering results of G_c . It reveals that with the increase of $D_s(s_1)$, achieving G_c requires an improved steering ability (see the value of $\max\{|\dot{\theta}|\}$ in the figure for example).

Table V summarizes the required angular speed limit and its averaged travel time \bar{t}_f for the three groups of paths. It has been found that there is no solution for too small δ ; for example, $\delta < 0.3^\circ/s$ for $D_s(s_1) \in [30 m, 60 m]$, and $\delta < 0.4^\circ/s$ for $D_s(s_1) \in [70 m, 90 m]$. The difference in minimum valid δ occurs in G_a when $D_s(s_1)$ changes from 60 m to 70 m, which results in the decrease of number of solutions, as well as a decreased \bar{t}_f . From G_a to G_c , it is clear that increasing δ can make \bar{t}_f decrease, which reveals the fact that vehicles with higher steering ability are more suited to close-range maneuvering.

TABLE V
SCOPE OF THRUST ANGULAR SPEED AND AVERAGED TRAVEL TIME FOR G_a , G_c AND G_c

$D_s(s_1)[m]$	G_a		G_b		G_c	
	$\delta[^\circ/s]$	$\bar{t}_f[s]$	$\delta[^\circ/s]$	$\bar{t}_f[s]$	$\delta[^\circ/s]$	$\bar{t}_f[s]$
30	[0.3,1.2]	553.16	(1.2,7.7)	482.49	(7.7,18]	479.42
40	[0.3,1.1]	616.72	(1.1,8.2]	502.05	(8.2,18]	497.18
50	[0.3,0.9]	823.81	(0.9,11.0]	528.29	(11.0,18]	521.18
60	[0.3,1.2]	840.92	(1.2,10.4]	553.83	(10.4,18]	549.77
70	[0.4,1.2]	781.47	(1.2,10.3]	588.41	(10.3,18]	583.56
80	[0.4,1.3]	790.71	(1.3,9.2]	627.44	(9.2,18]	623.37
90	[0.4,2.4]	733.15	(2.4,14.6]	670.70	(14.6,18]	669.39

TABLE VI
THE POSITIONS, ORIENTATIONS, AND VELOCITIES OF TSS

Dynamic obstacles	$X_d[m]$	$\Phi_d[^\circ]$	$V_d[m/s]$
TS ₁	(50, 200)	10	0.55
TS ₂	(125, 400)	-100	0.60
TS ₃	(425, 450)	-150	0.45
TS ₄	(200, 450)	-15	0.15
TS ₅	(450, 50)	180	0.40
TS ₆	(250, 200)	-30	0.65
TS ₇	(450, 350)	-90	0.20

D. Path Planning with Multiple Obstacles

The experiment was carried out to verify the possibility of collision avoidance among multiple close-range encounters. We continued to use the scene in Section V-C, including X_0 , X_{t_f} and D_0 of the OS, and the gyre parameters. Four static obstacles, with the same radius of $D_s(s_i)=30 m$ and clearance distance $C_s(s_i)=D_s(s_i)+11.5 m$, are located at $(200 m, 400 m)$, $(250 m, 290 m)$, $(300 m, 125 m)$ and $(375 m, 200 m)$, respectively. In addition, there are seven dynamic obstacles from TS₁ to TS₇ involved in the experiment, with their initial states listed in Table VI. Their dimensions are set the same as the OS. The optimization problem here is the same as (4). The related parameters for the OS and the TSs are set as $v=0.7 m/s$ and $\delta=18^\circ/s$, $T_0=0 s$, $C_{t_f}=20 m$, $C_d=90 m$, $p=1000$ and $q=10$, respectively. The computation time for this experiment

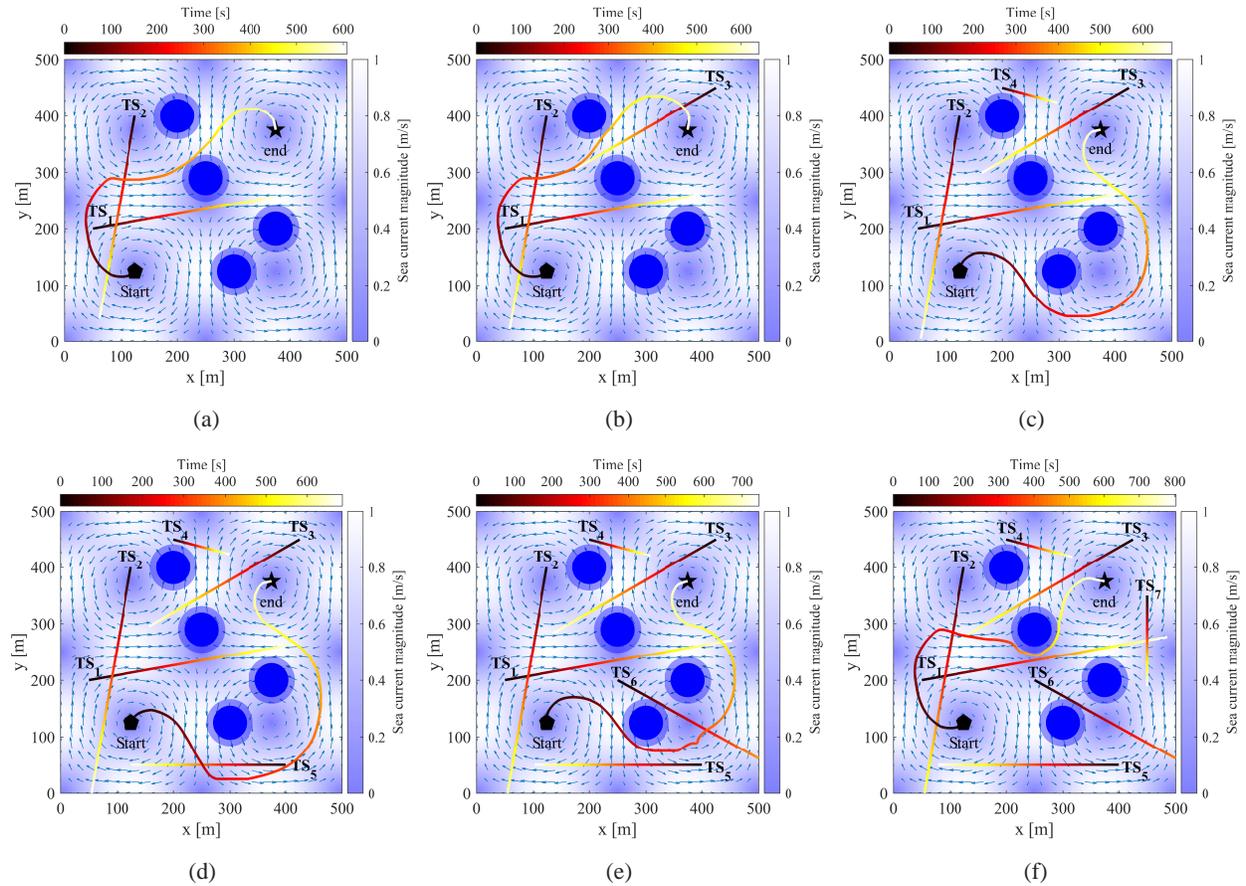


Fig. 10. Solutions for the “multiple obstacles” optimization problem with respect to different number of TSs. (a) 2 TSs. (b) 3 TSs. (c) 4 TSs. (d) 5 TSs. (e) 6 TSs. (f) 7 TSs.

increases from 144s to 187s with the growth of number of TSs.

Fig. 10 shows six test cases with a different number of TSs. Note that for each test case, an extra TS is added to arrange around the planned path from the previous test case. The purpose is to examine how these minimum-time paths evolve with the increase of TSs. In order to avoid collision with TSs, Fig. 10a, 10b, 10d, and 10e reveal that the planner tries to slightly change the OS’s course at the place where the sea’s current magnitude is relatively small. If changing course cannot guarantee the reachability or collision avoidance, a new path will be explored, as depicted in Fig. 10c and 10f.

There are head-on, crossing and overtaking between the OS and the TSs in Fig. 10. The experiment does not identify any instances of high risk of ES (see Fig. 5c). This is consistent with (17), where high risk of ES corresponds to high cost. Table VII lists the closest ES for each test case, together with the minimal distance and the time when it happens. According to

TABLE VII
CLOSEST ENCOUNTER SITUATIONS IN FIG. 10

Test case	Encounters	Min. dist. [m]	Time of min. dist.[s]	Encounter situation
2 TSs	OS & TS ₂	42.46	239.80	O_2T_2
3 TSs	OS & TS ₂	42.46	239.80	O_2T_2
	OS & TS ₃	32.27	422.70	O_2T_4
4 TSs	OS & TS ₁	49.21	549.51	O_4T_2
5 TSs	OS & TS ₁	45.88	570.65	O_4T_2
	OS & TS ₅	25.03	246.74	O_3T_4
6 TSs	OS & TS ₁	37.61	614.36	O_4T_2
	OS & TS ₅	27.98	243.29	O_2T_4
	OS & TS ₆	54.25	345.89	O_2T_2
7 TSs	OS & TS ₁	58.74	358.34	O_2T_1
	OS & TS ₂	42.46	239.80	O_2T_2
	OS & TS ₄	51.66	741.63	O_4T_2

Table II, the OS in these closest ESs is “stand-on”, which indicates the OS is safe even when the TS is within the circle of clearance distance.

We have also verified successful trajectory planning in other cases of X_0 and X_{tf} with non-zero sea current velocity. Considering the length of the paper, the results are not shown here. From the experiment, we demonstrate the effectiveness of the proposed planning method for collision avoidance in complicated close-range encounter scenarios.

E. Discussion

This section discusses the tuning of planning parameters, the key factors of maneuvering ability, and the compliance with COLREGs for close-range encounters in the complex spatial environment.

The proposed planner mainly contains three groups of adjustable parameters. The first group of parameters including m , n , Δl , L , Δc and $\Delta\alpha$ is used to model the sea current distribution. As illustrated in Fig. 3, these parameters determine the number of nodes and connections of the directed graph. An empirical setting of $m, n \in [50, 500]$, $L \in [\Delta l, 10\Delta l]$, $\Delta c \in (0, 0.2]$ and $\Delta\alpha \in (0^\circ, 4^\circ]$ would balance the computational complexity and the reachability for close-range maneuvering applications. The second group is user-defined parameters T_0 and C_{tf} used

for cost evaluation in initial and final states. From (19) – (21), these parameters should be kept in a relatively small and reasonable range, e.g., $T_0 \in (0\text{ s}, 10\text{ s}]$ and $C_{tf} \in (0\text{ m}, 30\text{ m}]$, to prevent the resultant trajectory from violating the initial and final state constraints. The last group is collision avoidance related parameters C_s and C_d , and their values depend on OS's and obstacles' dimensions. For ship safety, it is suggested to set C_s at least half the width of the OS wider than D_s , and set C_d longer than the sum of length and width of D_d , respectively.

Maneuvering ability refers to both v and δ according to (1) and (2). From Fig. 4, a higher value of v will have greater possibility of finding solutions, thereby improving reachability to neighbor nodes. This is consistent with practical applications and has also been verified in Fig. 8, in which the number of obtained paths increases when v increases. δ also has an effect on maneuvering. As illustrated in Table V, when v is relatively high, the influence of δ becomes more prominent. This makes sense, given that a higher value of v corresponds to a shorter travel time to reach available neighbors. From (10), however, the range of valid thrust orientation will be narrowed down accordingly. That is why, in most close-range maneuvering applications, using a high value of δ on the vehicle is preferable.

Regarding the compliance with COLREGs for the proposed planner, it is closely related to how the COLREGs are implemented. In the COLREGs, the “give-way” vehicle is suggested to take action as early as possible to keep the way clear. The strategy in (17) interprets the COLREGs in a similar way in that the OS keeps the TSs out of a safe area in case there is a high risk of collision. Fig. 8 and Table VII verify the correctness of the idea. The planner may not completely follow rule 6 in the COLREGs, that is, to guarantee the OS to pass by obstacles using a safe/low speed, due to the conflict with the optimization goal. The assumption of constant v in the kinematic model may be the other factor that makes it happen. Utilizing variable v is promising to address the problem, as at least it will be beneficial to deal with dense traffic scenarios. But a full understanding of the use of variable v is beyond the scope of this paper and will be investigated in future work.

To sum up, the proposed planner takes both maneuvering ability and the COLREGs into account and if the planning parameters are well tuned, it is capable of generating time efficient paths under various constraints, ranging from sea current, obstacles, and initial and final states, to thrust orientation.

VI. CONCLUSION

Intelligent trajectory planning is the crucial element of autonomous ships for the next generation of marine transportation systems. In this paper, we have investigated the time-optimal trajectory planning in close-range encounters. Considering that the surroundings of a surface vessel during maneuvering create complex spatial variability, including sea current, marine traffic, and constraints from initial and final states, a two-stage trajectory planning scheme is proposed. In the first stage, a directed graph with variable connection is established. The reachability, the risk of collision, and different states constraints determine the maneuvering cost. Through a wavefront search, a discrete solution can be obtained. The discrete path is parameterized as multiple Bézier curves connecting with C^2 continuity in the second stage. Searching in the parameter space via GA provides a way to explore the discrete path. By using the same cost criteria in the wavefront search, the solution will be a near-optimal but smooth path. Through a benchmark test and experiments with respect to maneuvering ability and COLREGs compliance, we confirm the effectiveness of the planner for generating a minimum-time path in close-range maneuvering.

Based on the current work and the discussion in Section V-E, future efforts will be made to (1) apply variable vehicle velocity to the kinematic model and refine the two-stage scheme to address the corresponding optimization problem; (2) use hierarchical approaches to improve the proposed planner and thus increase the re-planning frequency; and (3) combine the proposed planner with a trajectory tracking controller to achieve autonomous maneuvering in a professional simulation platform.

ACKNOWLEDGMENT

The research is supported by a grant from the Knowledge-Building Project for Industry “Digital Twins for Vessel Life Cycle Service” (Project nr: 280703), and by a grant from the Research-Based Innovation “SFI Marine Operation in Virtual Environment (SFI-MOVE)” (Project nr: 237929) in Norway.

REFERENCES

- [1] E. Jokioinen, J. Poikonen, M. Hyvonen, A. Kolu *et al.*, “Remote and autonomous ships-the next steps,” *London: AAWA Position Paper, Rolls-Royce*, 2016.
- [2] World’s first autodocking installation successfully tested by wärtsilä. [Online]. Available: <https://www.wartsila.com/media/news/26-04-2018-world-s-first-autodocking-installation-successfully-tested-by-wartsila-2169290>

- [3] New research project to investigate sensor fusion and collision avoidance for advanced ships. [Online]. Available: <https://www.km.kongsberg.com/ks/web/nokbg0238.nsf/AllWeb/2166175A7B81AE08C1257E58003CCE1B?OpenDocument>
- [4] T. I. Fossen, *Marine control systems: guidance, navigation and control of ships, rigs and underwater vehicles*. Trondheim, Norway: Marine Cybernetics, 2002.
- [5] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.
- [6] N. Ganganath, C.-T. Cheng, and K. T. Chi, "A constraint-aware heuristic path planner for finding energy-efficient paths on uneven terrains," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 601–611, 2015.
- [7] B. Li, K. Wang, and Z. Shao, "Time-optimal maneuver planning in automatic parallel parking using a simultaneous dynamic optimization approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 11, pp. 3263–3274, 2016.
- [8] A. Azaron and F. Kianfar, "Dynamic shortest path in stochastic dynamic networks: Ship routing problem," *European Journal of Operational Research*, vol. 144, no. 1, pp. 138–156, 2003.
- [9] J.-O. Kim and P. K. Khosla, "Real-time obstacle avoidance using harmonic potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 338–349, 1992.
- [10] S. Quinlan, "Real-time modification of collision-free paths," Ph.D. dissertation, Stanford University, 1994.
- [11] S. M. LaValle and J. J. Kuffner Jr, "Rapidly-exploring random trees: Progress and prospects," *Algorithmic and Computational Robotics: New Directions*, pp. 293–308, 2000.
- [12] D. Hsu, "Randomized single-query motion planning in expansive spaces," Ph.D. dissertation, Stanford University, 2000.
- [13] J. Y. Hwang, J. S. Kim, S. S. Lim, and K. H. Park, "A fast path planning by path graph optimization," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 33, no. 1, pp. 121–129, 2003.
- [14] B. Garau, A. Alvarez, and G. Oliver, "Path planning of autonomous underwater vehicles in current fields with complex spatial variability: an a* approach," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 194–198.
- [15] E.-i. Kobayashi, T. Asajima, and N. Sueyoshi, "Advanced navigation route optimization for an oceangoing vessel," *International Journal on Marine Navigation and Safety of Sea Transportation*, vol. 5, no. 3, pp. 377–383, 2011.
- [16] G. Li and H. Zhang, "A bézier curve based ship trajectory optimization for close-range maritime operations," in *ASME 2017 36th International Conference on Ocean, Offshore and Arctic Engineering*. American Society of Mechanical Engineers, 2017, pp. V07BT06A026–V07BT06A026.
- [17] M. Pivtoraiko, R. A. Knepper, and A. Kelly, "Differentially constrained mobile robot motion planning in state lattices," *Journal of Field Robotics*, vol. 26, no. 3, pp. 308–333, 2009.
- [18] B. C. Shah, P. Švec, I. R. Bertaska, A. J. Sinisterra, W. Klinger, K. von Ellenrieder, M. Dhanak, and S. K. Gupta, "Resolution-adaptive risk-aware trajectory planning for surface vehicles operating in congested civilian traffic," *Autonomous Robots*, vol. 40, no. 7, pp. 1139–1163, 2016.
- [19] Y. Kuwata, M. T. Wolf, D. Zarzhitsky, and T. L. Huntsberger, "Safe maritime autonomous navigation with colregs, using velocity obstacles," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 110–119, 2014.
- [20] A. Maki, Y. Akimoto, Y. Nagata, S. Kobayashi, E. Kobayashi, S. Shiotani, T. Ohsawa, and N. Umeda, "A new weather-routing system that accounts for ship stability based on a real-coded genetic algorithm," *Journal of marine science and technology*, vol. 16, no. 3, p. 311, 2011.
- [21] J.-J. Kim and J.-J. Lee, "Trajectory optimization with particle swarm optimization for manipulator motion planning," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 620–631, 2015.
- [22] B. Yoo and J. Kim, "Path optimization for marine vehicles in ocean currents using reinforcement learning," *Journal of Marine Science and Technology*, vol. 21, no. 2, pp. 334–343, 2016.

- [23] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa, "Chomp: Covariant hamiltonian optimization for motion planning," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1164–1193, 2013.
- [24] J. Dong, M. Mukadam, F. Dellaert, and B. Boots, "Motion planning as probabilistic inference using gaussian processes and factor graphs," *Proceedings of Robotics: Science and Systems (RSS-2016)*, 2016.
- [25] Y. T. Tan, R. Gao, and M. Chitre, "Cooperative path planning for range-only localization using a single moving beacon," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 2, pp. 371–385, 2014.
- [26] B. Rhoads, I. Mezić, and A. C. Poje, "Minimum time heading control of underpowered vehicles in time-varying ocean currents," *Ocean Engineering*, vol. 66, pp. 12–31, 2013.
- [27] A. L. Roald, "Path planning for vehicle motion control using numerical optimization methods," Master's thesis, NTNU, 2015.
- [28] A. M. Lekkas, A. L. Roald, and M. Breivik, "Online path planning for surface vehicles exposed to unknown ocean currents using pseudospectral optimal control," *IFAC-PapersOnLine*, vol. 49, no. 23, pp. 1–7, 2016.
- [29] L. Walther, A. Rizvanolli, M. Wendebourg, and C. Jahn, "Modeling and optimization algorithms in ship weather routing," *International Journal of e-Navigation and Maritime Economy*, vol. 4, pp. 31–45, 2016.
- [30] M. Soulignac, P. Taillibert, and M. Rueher, "Time-minimal path planning in dynamic current fields," in *IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 2473–2479.
- [31] Z. Zeng, K. Sammut, L. Lian, A. Lammas, F. He, and Y. Tang, "Rendezvous path planning for multiple autonomous marine vehicles," *IEEE Journal of Oceanic Engineering*, vol. 43, no. 3, pp. 640–664, 2018.
- [32] B. Rhoads, I. Mezić, and A. Poje, "Minimum time feedback control of autonomous underwater vehicles," in *2010 49th IEEE Conference on Decision and Control (CDC)*. IEEE, 2010, pp. 5828–5834.
- [33] T. Lolla, P. F. Lermusiaux, M. P. Ueckermann, and P. J. Haley Jr, "Time-optimal path planning in dynamic flows using level set equations: theory and schemes," *Ocean Dynamics*, vol. 64, no. 10, pp. 1373–1397, 2014.
- [34] U. Commandant, "International regulations for prevention of collisions at sea, 1972 (72 colregs)," *US Department of Transportation, US Coast Guard, COMMANDANT INSTRUCTION M*, vol. 16672, 1999.
- [35] W. Naeem, G. W. Irwin, and A. Yang, "Colregs-based collision avoidance strategies for unmanned surface vehicles," *Mechatronics*, vol. 22, no. 6, pp. 669–678, 2012.
- [36] C. Tam and R. Bucknall, "Path-planning algorithm for ships in close-range encounters," *Journal of marine science and technology*, vol. 15, no. 4, pp. 395–407, 2010.
- [37] T. A. Johansen, T. Perez, and A. Cristofaro, "Ship collision avoidance and colregs compliance using simulation-based control behavior selection with predictive hazard assessment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3407–3422, 2016.
- [38] H.-T. L. Chiang and L. Tapia, "Colreg-rrt: An rrt-based colregs-compliant motion planner for surface vehicle navigation," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2024–2031, 2018.
- [39] H. K. Lo and M. R. McCord, "Routing through dynamic ocean currents: General heuristics and empirical results in the gulf stream region," *Transportation Research Part B: Methodological*, vol. 29, no. 2, pp. 109–124, 1995.
- [40] R. Takei, R. Tsai, H. Shen, and Y. Landa, "A practical path-planning algorithm for a simple car: a hamilton-jacobi approach," in *American Control Conference (ACC), 2010*. IEEE, 2010, pp. 6175–6180.
- [41] C. Tam and R. Bucknall, "Collision risk assessment for ships," *Journal of marine science and technology*, vol. 15, no. 3, pp. 257–270, 2010.
- [42] E. Haines, "Point in polygon strategies," *Graphics gems IV*, vol. 994, pp. 24–26, 1994.

- [43] R. Bartels, J. Beatty, and B. Barsky, "Bézier curves," *An Introduction to Splines for Use in Computer Graphics and Geometric Modelling*, pp. 211–245, 1998.
- [44] B. A. Barsky and T. D. DeRose, "Geometric continuity of parametric curves: three equivalent characterizations," *IEEE Computer Graphics and Applications*, vol. 9, no. 6, pp. 60–69, 1989.
- [45] J.-W. Choi, R. Curry, and G. Elkaim, "Piecewise bezier curves path planning with continuous curvature constraint for autonomous driving," in *Machine learning and systems engineering*. Springer, 2010, pp. 31–45.
- [46] A. E. Bryson and Y.-C. Ho, *Applied optimal control: optimization, estimation, and control*. Washington : Hemisphere Pub. Corp., 1975.



Guoyuan Li (M'14) received the Ph.D. degree from the Institute of Technical Aspects of Multimodal Systems (TAMS), Department of Informatics, University of Hamburg, Hamburg, Germany, in 2013. From 2014, he joined the Mechatronics Laboratory, Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, Norway. His research interests include path planning, ship motion prediction, maneuvering control, artificial intelligence, optimization algorithms and locomotion control of bio-inspired robots. In these areas, he has published over 40 papers.



Hans Petter Hildre is professor and head of the Department of Ocean Operations and Civil Engineering at the Norwegian University of Science and Technology (NTNU). His area of interest is product design and system architecture design. Hans Petter is Centre Director for Centre for Research Driven Innovation (SFI-MOVE) within marine operations. This is cooperation between NTNU, SINTEF, University Sao Paulo and 15 companies at the west coast of Norway. Professor Hildre is head of research in national program Global Centre of Expertise Blue Maritime, project leader in several research projects, member of the board in 5 companies, and has a number of patents.



Houxiang Zhang (M'04-SM'12) received Ph.D. degree in Mechanical and Electronic Engineering in 2003. From 2004, he worked as Postdoctoral Fellow at the Institute of Technical Aspects of Multimodal Systems (TAMS), Department of Informatics, Faculty of Mathematics, Informatics and Natural Sciences, University of Hamburg, Germany. In Feb. 2011, he finished the Habilitation on Informatics at University of Hamburg. Dr. Zhang joined the NTNU, Norway in April 2011 where he is a Professor on Robotics and Cybernetics. The focus of his research lies on two areas. One is on biological robots and modular robotics. The second focus is on virtual prototyping and maritime mechatronics. In these areas, he has published over 130 journal and conference papers and book chapters as author or co-author.