Reachability-Based Planning of Time-Optimal Curvature-Constrained Path With Moving and Deforming Obstacles

Y. M. H. Xiao, H. Wang, and Y. Pan

***Abstract*— In this work, we develop a time-optimal path planning algorithm for a mobile robot constrained by a minimum turning radius in an environment cluttered with an arbitrary number of moving and deforming obstacles. The algorithm builds on our previous work and involves substantial extensions to handle the turning radius constraint by adding the heading angle of the robot to the state space in addition to its location in a 2-D plane. The developed planner involves two stages: 1) forward propagation of the reachable set in the state space to preset destination through a newly derived variational inequality (VI) which encodes the obstacle avoidance and 2) backtracking to obtain the waypoints of the optimal path (corresponding to optimal control of the turning rate and speed), solved through an ODE-based scheme or a new and more robust backward-set-based scheme. The planned path represents a rigorous global optimal solution (except numerical errors) to the problem that can be used as a benchmark for other simplified planners or implemented together with a receding horizon for path planning with limited perception ability. We demonstrate both applications in several test cases.**

***Index Terms*— Obstacle avoidance, path planning, reachability analysis.**

I. Introduction

THE past few decades have witnessed increasing deployments of mobile robots (which, broadly speaking, includes autonomous transportation vehicles) in both military and civilian applications, due to their potential to significantly improve mission efficiency and safety. Among the challenges associated with the integrated technology, a key component is the algorithm of optimal path planning which enables safe and efficient operation in a dynamic environment cluttered with various obstacles of different natures [1], [2], [3].

The development of path planning algorithms starts from an early age with typical methods utilizing graph search algorithm [4], [5], rapidly exploring random tree [6], artificial potential field [7], [8], and genetic algorithm [9]. This field of research has been very dynamic and rich, considering the various objectives and contexts of performing optimal path planning. In this work, we consider specifically the problem of time-optimal path planning of a mobile robot with a minimum turning radius in an environment cluttered with an arbitrary number of moving and deforming obstacles. This problem is essential in practical situations of time-sensitive missions (e.g., rescue, retreat from a natural disaster) in a hostile and time-varying environment (e.g., unmanned aerial vehicles operating in a region cluttered with hazardous zones contaminated by chemicals, unmanned surface vessels in a harsh sea with emerging, propagating, and deforming rogue waves). One critical challenge for the robot to avoid collision with moving and deforming obstacles is caused by the difficulty to obtain a well-defined obstacle velocity that is necessary for most existing avoidance algorithms [10], [11], [12]. To the authors' knowledge, there is so far no algorithm that can provide a rigorous global solution to such a fundamental problem, in spite of some methods partially related to the problem that we review below.

Matveev et al. [12] developed a purely reactive algorithm for mobile robots operating in an environment with moving and deforming obstacles. The purpose of the algorithm is to control the heading of the mobile robots for obstacle avoidance while minimizing the deviation from the desired heading. Although the avoidance can be proved rigorously, the algorithm cannot handle the kinematic constraint of the robot's minimal turning radius and complex obstacles which split into parts or merge together. Neither is the algorithm suitable for path planning problems with a given target destination, since the robot can drift arbitrarily from the desired path.

Another category of algorithm, which may be more appropriate in finding global optimal solutions with given starting and target points, is based on the Hamilton–Jacobi reachability theory. The key idea of the theory when applied to the path planning problem is to: 1) track the propagation of the forward reachable set (i.e., the set of all states that a mobile robot can reach at time t) from the initial set (as the starting state) until it reaches the target state and 2) backtrack to obtain the optimal state trajectory (and its corresponding optimal control). Following this idea, lolla et al. [13] and Lermusiaux et al. [14]. developed a time-optimal path planner for vessels operating in dynamic ocean currents with moving obstacles. However, the method is not directly applicable to deforming obstacles

Manuscript received 4 September 2023; revised 1 June 2024; accepted 8 June 2024. Date of publication 2 July 2024; date of current version 23 October 2024. The work of Y. M. H. Xiao and H. Wang was supported by the National Natural Science Foundation of China under Grant 51909162. *(Corresponding author: Y. Pan.)*

Y. M. H. Xiao and H. Wang are with the School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: yuminghx@sjtu.edu.cn; whd302@sjtu.edu.cn).

Y. Pan is with the Department of Naval Architecture and Marine Engineering, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: yulinpan@umich.edu).

Digital Object Identifier 10.1109/TCST.2024.3415377

and kinematic constraint of minimum turning radius, both of which are desired in our path planning problem.

One fundamental difficulty in handling deforming obstacles is that the traditional Hamilton–Jacobi–Bellman (HJB) and other methods reviewed above require the moving velocity of the obstacle as the input, which cannot be uniquely and well defined if the obstacle deforms, splits, or merges. For example, an oceanic rogue wave can emerge from nowhere, defying any valid velocity definition at the instant of their formation. Practically, it is also much easier to measure the field where obstacles (say some "dangerous regions") are embedded than the obstacle velocity considering the much higher uncertainty involved in the latter [15]. The measured field (e.g., surface elevation of the ocean) at current time can then be used as the initial condition for physical simulations to predict the field and obstacle locations at later time. To build a path planning algorithm requiring only the locations of the obstacles (so that applicable to deforming ones), our previous work [16] proposed a reachability-based time-optimal path planner, employing a variational inequality (VI) [17] for the propagation of the forward reachable set. The VI can be considered as a HJB equation corrected by the location of obstacles, which rigorously guarantees the obstacle avoidance and time optimality without using the velocity information of obstacles.

In this article, we further enrich our VI-based method to incorporate the kinematic constraint of minimal turning radius of the robot, rendering it more suitable for realistic applications. The development involves substantial extension to [16] as the full algorithm has to be re-formulated for a new augmented state space including the heading angle θ in addition to the location (x, y). To this end, we derive and numerically solve a new VI governing the evolution of the reachable set in the $[x \;\; y \;\; \theta]^T$ space for the forward propagation problem. Besides, we provide a rigors proof for the proposed VI in the framework of viscosity solution, which constitutes the main theoretical contribution of this work. Furthermore, we newly propose a backtracking scheme based on the local backward reachable set, which is more robust and accurate than the traditional ODE-based method. The full algorithm provides a rigorous global optimal solution (except numerical errors) to the path planning problem, which can be used as a verification algorithm to benchmark the optimality of other simplified path planners. In addition, we also apply the proposed reachability-based planning in a receding horizon manner, which, combined with implementations on a high-performance computer (e.g., with graphics processing unit (GPU) acceleration), can be used in real-time planning in case of limited perception ability. We demonstrate the validity of the method in several test cases including the time-optimal path planning of an autonomous surface vessel in a modeled ocean wave field with dangerous high waves treated as moving and deforming obstacles.

The remainder of this article is organized as follows. In Section II, we present the mathematical framework including a general problem statement and formulation based on the reachability theory. In Section III, we elaborate the main algorithm to solve the forward propagation and backtracking problems, with the receding horizon implementation discussed in Section IV. The validation cases are conducted in Section V, followed by a conclusion of the work in Section VI.

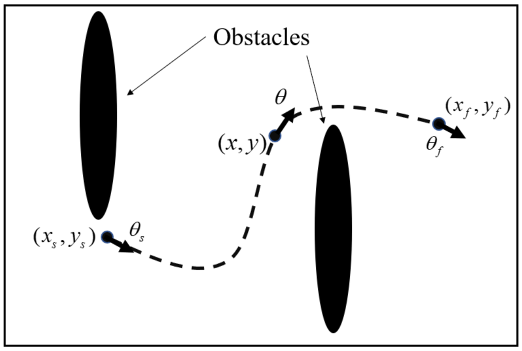


Fig. 1. Sketch of the path planning problem, including the initial state $[x_s \;\; y_s \;\; \theta_s]^T$, target state $[x_f \;\; y_f \;\; \theta_f]^T$, a representative state trajectory $\mathbf{P}(t)$ (– – –), and obstacles (black shadow).

II. Mathematical Framework

A. Problem Statement

A graphical description of the problem is sketched in Fig. 1, where a mobile robot is considered as a point mass moving in a 2-D plane cluttered with (moving and deforming) obstacles. We remark that the obstacles can be inflated appropriately should the size of the robot needs to be taken into account.

The configuration of the mobile robot can be fully described by a state vector $\mathbf{x} = [x \;\; y \;\; \theta]^T$, where (x, y) denotes the location and θ represents the heading angle, i.e., the angle of the robot heading direction in an Earth-fixed coordinate system. The state trajectory of the mobile robot $\mathbf{P}(t)$ is subjected to the following kinematic relation:

$$\dot{\mathbf{P}}(t) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} F(t) \cdot \cos\theta \\ F(t) \cdot \sin\theta \\ \omega(t) \end{bmatrix} \tag{1}$$

where $F(t)$ and $\omega(t)$ are the instantaneous speed and turning rate of the robot. We consider practical kinematic constraints of a maximum speed $F_{\max}$ and a minimum turning radius $R_{\min}$, so that the control inputs $F(t)$ and $\omega(t)$ fall into the subsequent constraint set as

$$\mathcal{U} \equiv \left\{ F(t) \in [0, F_{\max}],\ \omega(t) \in \left[-\frac{F(t)}{R_{\min}}, +\frac{F(t)}{R_{\min}} \right] \right\}. \tag{2}$$

Let $t_s = 0$ be the initial time, and t_f be the arrival time to be minimized. For a mobile robot starting from a given initial location (x_s, y_s) with a given heading θ_s to arrive at a preset target location (x_f, y_f), we have

$$\mathbf{P}(0) = \mathbf{x}_s, \quad \mathbf{P}(t_f) = \mathbf{x}_f \tag{3}$$

where $\mathbf{x}_s = [x_s \;\; y_s \;\; \theta_s]^T$ and $\mathbf{x}_f = [x_f \;\; y_f \;\; \theta_f]^T$ are the initial and target states of the mobile robot. We note that θ_f can be prescribed or set as arbitrary depending on the mission of the mobile robot. In this article, we focus on the latter case with arbitrary heading $\theta_f \in [0, 2\pi]$ [i.e., the robot is allowed to arrive at (x_f, y_f) with any heading angle] although the proposed algorithm can handle both cases.

We assume that the locations of moving and deforming obstacles can be predicted by certain physical computations, and they are known for current and future time. In an environment cluttered with such obstacles, the mobile robot is required not to enter the regions occupied by obstacles at any time. Hence, the state trajectory $\mathbf{P}(t)$ needs to satisfy

$$\mathbf{P}(t) \notin A_{\mathrm{obst}}(t) \quad \forall t \in [0, t_f] \tag{4}$$

where $A_{\mathrm{obst}}(t)$ denotes the set of restricted states due to obstacles. More specifically, $A_{\mathrm{obst}}(t)$ represents volumes in the $[x\ y\ \theta]^T$ state space whose projection on the xy plane gives the areas occupied by the obstacles, that is

$$\begin{aligned} A_{\mathrm{obst}}(t) \equiv \{\mathbf{x} = [x_{\mathrm{obst}}\ y_{\mathrm{obst}}\ \theta]^T\ &|(x_{\mathrm{obst}}, y_{\mathrm{obst}}) \in \text{areas} \\ &\text{occupied by the obstacle at } t,\ \theta \in [0, 2\pi]\}. \end{aligned} \tag{5}$$

Under the terminology of reachability theory, a state trajectory $\mathbf{P}(t)$ is called admissible if it satisfies (1) and (4) with the corresponding controls $F(t)$, $\omega(t)$ constrained by $\mathcal{U}$.

The objective of our time-optimal path planner is to find

$$t_f^* = \operatorname*{Inf}_{\omega(t), F(t)\ \in \mathcal{U}} \{t_f\} \tag{6}$$

with constraints (1)–(4), as well as the optimal state trajectory $\mathbf{P}^*(t)$ resulting from the optimal controls $F^*(t)$ and $\omega^*(t)$.

B. Reachability Theory Applied to Path Planning

The key of applying the reachability theory to the proposed path planning problem is the computation of the forward reachable set, which is defined as the set of all states that the mobile robot can reach at time t

$$\begin{aligned} R_0[t] \equiv \{\mathbf{x} \in \mathbb{R}^3\ |\mathbf{x} = \mathbf{P}(t),\ \exists\, \omega(\tau), F(\tau) \underset{\tau\in[0,t]}{\in} \mathcal{U} \text{ such that:} \\ 1)\ \mathbf{P}(\tau) \text{ admissible},\ 2)\ \mathbf{P}(0) = \mathbf{x}_s\}. \end{aligned} \tag{7}$$

According to (7), for any $\mathbf{x}$ in $R_0[t]$, there must exist control inputs that give an admissible trajectory connecting $\mathbf{x}_s$ and $\mathbf{x}$ (see Fig. 2(a) for a graphical description of the concepts). The boundary of $R_0[t]$ is termed as the forward reachable front $\partial R_0[t]$ (i.e., the "farthest" set that can be reached by the mobile robot in the state space at time t).

The minimal arrival time t_f^* [as in (6)] can be determined as the time when $\partial R_0[t]$ first reaches the target state $\mathbf{x}_f$ (which is intuitively true with its proof provided in [13]). The state trajectory $\mathbf{P}(t)$ which stays on $\partial R_0[t]$ (i.e., $\mathbf{P}(t) \in \partial R_0[t]$ for $t \in [0, t_f^*]$) and connects $\mathbf{x}_s$ and $\mathbf{x}_f$ is the $\mathbf{P}^*(t)$ corresponding to the optimal controls. The projection of $\mathbf{P}^*(t)$ onto the xy plane provides the time-optimal path for the mobile robot (see Fig. 2(b) for an illustration).

Accordingly, two steps are needed in our path planning algorithm: 1) forward propagation of $R_0[t]$ until $\partial R_0[t]$ reaches $\mathbf{x}_f$ at t_f^* and 2) backtracking to find optimal intermediate waypoints $\mathbf{P}^*(t_n)$ on $\partial R_0[t_n]$ ($t_n = n\Delta t$, Δt being the size of the time step) as the optimal state trajectory $\mathbf{P}^*(t)$. Intuitively, this algorithm guarantees that an optimal solution can be found if there exist admissible paths connecting $\mathbf{x}_s$ and $\mathbf{x}_f$. On the other hand, if the constrained kinematics (1) is not sufficient to evade some obstacles, $R_0[t]$ collapses to the boundary of $A_{\mathrm{obst}}(t)$, and a destined collision can be predicted. We focus on the first case with the existence of paths in this article.

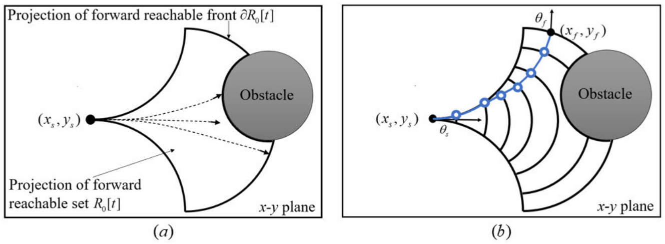


Fig. 2. Illustration of concepts of reachability-based path planning with the kinematic constraint of a minimum turning radius. (a) Reachable set $R_0[t]$, reachable front $\partial R_0[t]$, admissible paths (– – –) projected onto the xy plane. (b) Optimal state trajectory $\mathbf{P}^*(t)$ (——) with $\mathbf{P}^*(t_n) \in \partial R_0[t_n]$ (○) projected onto the xy plane.

III. Methodology

In this section, we first review the path planning algorithm for mobile robots without the kinematic constraint of R_{min}, i.e., the case solved in [16]. This review serves as a mild preparation for the cases with the constraint of R_{min}, for which we formulate the algorithm in detail afterward.

A. Path Planning Without the R_{min} Kinematic Constraint

If the R_{min} kinematic constraint is not considered, the heading angle is not needed in the state space, and all the reachability-based concepts [e.g., $R_0[t]$, $\partial R_0[t]$, A_{obst}, and $\mathbf{P}^*(t)$] can be defined on a 2-D state space $\mathbf{x}^{\mathrm{loc}} = [x\ \ y]^T$ (see Fig. 3(a) for an illustration of the corresponding $R_0^{\mathrm{loc}}[t]$, $\partial R_0^{\mathrm{loc}}[t]$, $A_{\mathrm{obst}}^{\mathrm{loc}}$, and $\mathbf{P}_{\mathrm{loc}}^*(t)$]. The kinematic relation (1) is also reduced to a simpler form

$$\dot{\mathbf{P}}_{\mathrm{loc}}(t) = F(t) \cdot \mathbf{h}(t) \tag{8}$$

with control inputs $F(t) \in [0, F_{\mathrm{max}}]$, and $\mathbf{h}(t)$ being a unit vector in the heading direction.

The forward propagation of $R_0^{\mathrm{loc}}[t]$ is then solved through a level set method, where $R_0^{\mathrm{loc}}[t]$ is represented as the nonpositive part of a value function $\phi(\mathbf{x}^{\mathrm{loc}}, t)$, i.e., $R_0^{\mathrm{loc}}[t] = \{\mathbf{x}^{\mathrm{loc}} \in \mathbb{R}^2 | \phi(\mathbf{x}^{\mathrm{loc}}, t) \le 0\}$ [see Fig. 3(c)]. We further encode $A_{\mathrm{obst}}^{\mathrm{loc}}(t)$ in another value function $g(\mathbf{x}^{\mathrm{loc}}, t)$ (which is defined as $A_{\mathrm{obst}}^{\mathrm{loc}}(t) = \{\mathbf{x}^{\mathrm{loc}} | g(\mathbf{x}^{\mathrm{loc}}, t) > 0\}$ through the known information about the obstacles). As in [16] and [17], the evolution of $\phi(\mathbf{x}^{\mathrm{loc}}, t)$ is supposed to be governed by a HJB type VI, taking the form of

$$\min\left\{\phi_t + H(\mathbf{x}^{\mathrm{loc}}, t, \nabla\phi), \phi(\mathbf{x}^{\mathrm{loc}}, t) - g(\mathbf{x}^{\mathrm{loc}}, t)\right\} = 0 \tag{9}$$

with the Hamiltonian here defined as

$$H(\mathbf{x}^{\mathrm{loc}}, t, \nabla\phi) = \sup_{\substack{F(t)\in[0, F_{\mathrm{max}}] \\ \mathbf{h}(t) \text{ a unit vector}}} \left\{\nabla\phi \cdot \dot{\mathbf{P}}_{\mathrm{loc}}(t)\right\} = |\nabla\phi| \cdot F_{\mathrm{max}} \tag{10}$$

where the controls $F(t)$ and $\mathbf{h}(t)$ aspire to maximize the Hamiltonian and lead to the second equivalence.

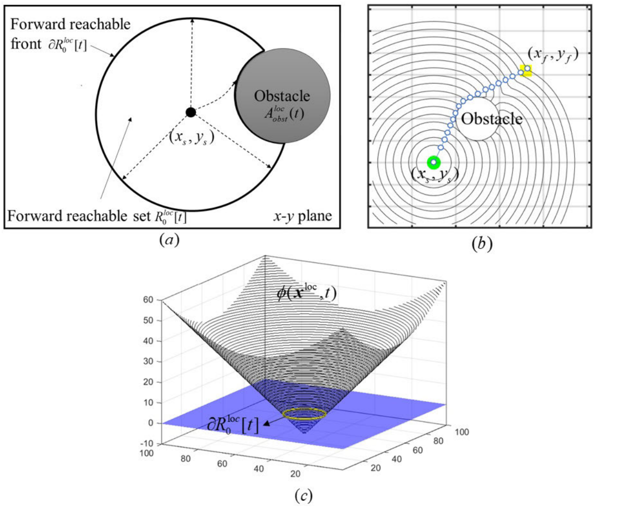

Forward reachable front $\partial R_0^{loc}[t]$
Obstacle $A_{obst}^{loc}(t)$
(x_s, y_s)
Forward reachable set $R_0^{loc}[t]$
x-y plane
(a)
(x_f, y_f)
Obstacle
(x_s, y_s)
(b)
$\phi(\mathbf{x}^{loc}, t)$
$\partial R_0^{loc}[t]$
60
50
40
30
20
10
0
-10
100
80
60
40
20
20
40
60
80
100
(c)

Fig. 3. Illustration of the reachability-based path planner in the case without the $R_{\min}$ kinematic constraint. (a) Definition of reachable set $R_0^{\text{loc}}[t]$ and relevant concepts. (b) Backtracking of $\mathbf{P}^*_{\text{loc}}(t)$ from (x_f, y_f) to (x_s, y_s). (c) Reachable set $R_0^{\text{loc}}[t]$ represented by the value function $\phi(\mathbf{x}^{\text{loc}}, t)$.

Next, to prove that the value function $\phi(\mathbf{x}^{\text{loc}}, t)$ actually solves the VI (9), we can study it in the viscosity solution framework, which allows for more general analysis of nonsmooth solutions [18]. More specifically, the conditions for the value function $\phi(\mathbf{x}^{\text{loc}}, t)$ to be the viscosity solution of VI (9) are formulated according to its mathematical form. Then, one may check whether $\phi(\mathbf{x}^{\text{loc}}, t)$ satisfies such conditions (i.e., subsolution and supersolution properties) and employ the comparison principle to inspect the uniqueness of the solution. For detailed derivation of (9) following the above steps, the interested readers are referred to [16] and references therein, and we will elaborate the formulation of VI for the case with $R_{\min}$ kinematic constraint according to the same framework later in this section.

Given the evolution of $\phi(\mathbf{x}^{\text{loc}}, t)$ (thus $R_0^{\text{loc}}[t]$), an optimal path $\mathbf{P}^*_{\text{loc}}(t)$ can be planned through the backtracking scheme, which tracks $\mathbf{P}^*_{\text{loc}}(t)$ from (x_f, y_f) to (x_s, y_s) [see Fig. 3(b)]. This can be accomplished through the standard ODE-based method [14], [20] or a more robust backward-set-based method explained later.

B. Path Planning With the R_{min} Kinematic Constraint

With the $R_{\min}$ kinematic constraint, we describe the configuration of the mobile robot in a state space $\mathbf{x} = [x\ y\ \theta]^T$ and follow the notations and concepts presented in Section II. The formulation of the forward propagation problem under the framework of level set method involves two key components: 1) derivation of the governing equation (which is a VI) for a value function $\phi(\mathbf{x}, t)$ that describes $R_0[t]$ and 2) design of the numerical scheme to solve the VI, which are presented, respectively, in parts 1) and 2) in this section. We will then introduce the backtracking scheme to find the optimal waypoints $\mathbf{P}^*(t_n)$ in part 3).

1) Derivation of the Governing Equation: Similar as in Section III-A, we need two value functions $\phi(\mathbf{x}, t)$ and $g(\mathbf{x}, t)$, both $\mathbb{R}^3 \times [0, +\infty) \to \mathbb{R}$, to track $R_0[t]$ and $A_{\text{obst}}(t)$, respectively. Since the location of $A_{\text{obst}}(t)$ is known at each time instant, $g(\mathbf{x}, t)$ can be conveniently chosen as a Lipschitz-continuous signed distance function with respect to the boundaries of $A_{\text{obst}}(t)$ satisfying

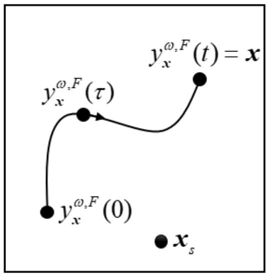

$y_{\mathbf{x}}^{\omega,F}(t) = \mathbf{x}$
$y_{\mathbf{x}}^{\omega,F}(\tau)$
$y_{\mathbf{x}}^{\omega,F}(0)$
$\mathbf{x}_s$

Fig. 4. Sketch for the notation $\mathbf{y}_{\mathbf{x}}^{\omega,F}(\tau)$, $\tau \in [0, t]$.

$$\begin{cases} g(\mathbf{x}, t) < 0, & \text{if } \mathbf{x} \notin A_{\text{obst}}(t) \\ g(\mathbf{x}, t) \geq 0, & \text{if } \mathbf{x} \in A_{\text{obst}}(t) \end{cases} \tag{11}$$

with $g(\mathbf{x}, t) = 0$ giving the boundaries of $A_{\text{obst}}(t)$.

The value function $\phi(\mathbf{x}, t)$ can be defined as

$$\phi(\mathbf{x}, t) \equiv \operatorname*{Inf}_{\substack{\omega(\tau), F(\tau) \in \mathcal{U} \\ \tau \in [0,t]}} \left\{ \max\left(\phi(\mathbf{y}_{\mathbf{x}}^{\omega,F}(0), 0), \max_{\tau} g(\mathbf{y}_{\mathbf{x}}^{\omega,F}(\tau), \tau) \right) \right\} \tag{12}$$

and

$$\begin{cases} \phi(\mathbf{x}, 0) \leq 0, & \text{if } \mathbf{x} \in R_0[0] \\ \phi(\mathbf{x}, 0) > 0, & \text{otherwise} \end{cases} \tag{13}$$

where $\mathbf{y}_{\mathbf{x}}^{\omega,F}(\tau)$, $\tau \in [0, t]$ is a state trajectory with endpoint $\mathbf{y}_{\mathbf{x}}^{\omega,F}(t) = \mathbf{x}$ corresponding to the control inputs $F(\tau)$ and $\omega(\tau)$ (see Fig. 4). $R_0[0]$ is the initial reachable set at $t = 0$ that can be prescribed in general. For our purpose of starting from a single state $\mathbf{x}_s$, it is preferable to prescribe (and think of) $R_0[0]$ as an infinitesimal volume $\mathbf{X}_s$ enclosing $\mathbf{x}_s$ [26]. This procedure is favorable for the robustness of both theoretical development (to allow $\phi(\mathbf{x}, 0)$ to be negative inside $\mathbf{X}_s$) and numerical implementation (to avoid the singularity at $\mathbf{x}_s$).

Equation (12) can be understood as follows: the first term about ϕ, with the Inf operation, evaluates whether there exist controls $\omega(\tau)$ and $F(\tau)$ resulting in a trajectory from $\mathbf{x}_s$ to $\mathbf{x}$. The second term about g serves as a penalty by obstacles, i.e., if the trajectory $\mathbf{y}_{\mathbf{x}}^{\omega,F}(\tau)$ encounters an obstacle, it is guaranteed that $\phi(\mathbf{x}, t) > 0$. Therefore, the nonpositive part of $\phi(\mathbf{x}, t)$ can be utilized to characterize $R_0[t]$, and that is to say

$$\begin{cases} \phi(\mathbf{x}, t) \leq 0, & \text{if } \mathbf{x} \in R_0[t] \\ \phi(\mathbf{x}, t) > 0, & \text{if } \mathbf{x} \notin R_0[t] \end{cases} \tag{14}$$

with $\partial R_0[t] = \{\mathbf{x} \in R^3 \,|\phi(\mathbf{x}, t) = 0\}$.

Based on the definition (12), if $g(\mathbf{x}, t)$ is not considered (i.e., obstacles do not exist), it is well established that $\phi(\mathbf{x}, t)$ satisfies the celebrated HJB equation

$$\phi_t + H(\mathbf{x}, t, \nabla\phi) = 0 \tag{15}$$

which can be derived through the dynamic programming principle [21]. Bringing in (1), the Hamiltonian is now defined as

$$\begin{aligned}H(\mathbf{x},t,\nabla\phi) &= \sup_{\omega(t),F(t)\in\mathcal{U}}\left\{\nabla\phi\cdot\dot{\mathbf{P}}(t)\right\}\\ &= \sup_{\omega(t),F(t)\in\mathcal{U}}\{F(t)\phi_x\cos\theta+F(t)\phi_y\sin\theta+\phi_\theta\omega(t)\}.\end{aligned}\tag{16}$$

After the maximization operation through choosing appropriate $F(t)$ and $\omega(t)$ (see Appendix A for details), (15) is reduced to

$$\phi_t+\max\left(0,\,F_{\max}\phi_x\cos\theta+F_{\max}\phi_y\sin\theta+|\phi_\theta|\frac{F_{\max}}{R_{\min}}\right)=0.\tag{17}$$

If $g(\mathbf{x},t)$ (i.e., obstacles) is considered, the function $\phi(\mathbf{x},t)$ with definition (12) can be shown to satisfy a VI in the following equation:

$$\min\left\{\underbrace{\phi_t+\max\left(0,\,\phi_x F_{\max}\cos\theta+\phi_y F_{\max}\sin\theta+|\phi_\theta|\,\frac{F_{\max}}{R_{\min}}\right)}_{\text{Term I}},\ \underbrace{\phi(\mathbf{x},t)-g(\mathbf{x},t)}_{\text{Term II}}\right\}=0\tag{18}$$

with initial condition (13). We summarize the proof for (18) in Appendix B, which builds on [17], [18], and [19] but includes important updates to correct the general proof procedure. It can be understood from (18) that the evolution of $\phi(\mathbf{x},t)$ is governed by either term I or term II set to zero, with the other term resulting in a nonnegative value. Intuitively, (17) implies that: 1) the HJB equation (expressed by term I $=0$) is only valid as the condition for optimality within regions denoted by term II ≥ 0 and 2) otherwise, the optimality condition can be expressed by term II $=0$ and term I ≥ 0. Practically, this process can be realized by solving term I $=0$ and correcting the resulted value function by term II, which is the essence of our numerical scheme discussed next.

2) Numerical Scheme: To solve (18) with (13) numerically, we propose a prediction-correction scheme, where two procedures are applied to march (18) in time.

1) *Prediction:* Evolve $I=0$ in (18) from t_n to t_{n+1} via

$$\hat{\phi}(\mathbf{x},t_{n+1})=\mathcal{F}(\phi(\mathbf{x},t_n))\tag{19}$$

where $\mathcal{F}$ is an operator depending on the numerical scheme.

2) *Correction:* Assign the maximum between $\hat{\phi}(\mathbf{x},t_{n+1})$ and $g(\mathbf{x},t_{n+1})$ to $\phi(\mathbf{x},t_{n+1})$, that is

$$\phi(\mathbf{x},t_{n+1})=\max\big(\hat{\phi}(\mathbf{x},t_{n+1}),g(\mathbf{x},t_{n+1})\big).\tag{20}$$

The above two steps (19) and (20) are executed iteratively until $\phi(\mathbf{x}_f,t)\leq 0$, i.e., $\partial R_0[t]$ first reaches $\mathbf{x}_f$ or more specifically, the projection of $\partial R_0[t]$ on the xy plane first reaches (x_f,y_f) with the robot taking any heading angle θ_f. The arrival time t of this instant corresponds to the optimal time t_f^* in (6).

To establish the operator $\mathcal{F}$, we discretize $I=0$ in (18) using a finite difference scheme. In particular, we apply a combination of the second-order essentially nonoscillatory (ENO2) scheme and the Godunov dissipation scheme to estimate the terms ϕ_x, ϕ_y, and ϕ_θ. As shown in [22] and [23], the ENO-Godunov scheme can avoid spurious oscillations in unsmooth regions of the value function and include proper numerical dissipation to keep the stability of the computation. For time integrator in (19), we use a total variation diminishing second-order Runge–Kutta (TVD-RK2) scheme with the time step determined by the Courant–Friedrichs–Lewy (C-F-L) condition [24], [25]. The details of the selected numerical schemes in this article are included in Appendix C.

What is more, since the discretized computational domain for solving (18) is finite, every dimension of the computational domain should be assigned with a specific boundary condition which requires an extension beyond the finite domain to obtain the derivatives at the boundary. Specifically, a certain number of ghost grids are appended to the upper and lower ends of the data array in each dimension, and the values filled in the ghost grids are dictated by the characteristic of the state variable and the type of the boundary condition. For our algorithm, a periodic boundary condition is applied in θ-direction for $\theta\in[0,2\pi]$, which copies data from the upper side of the array (i.e., 2π end) to the lower ghost grids (i.e., 0 end), and vice versa. As for x and y dimensions, an open boundary condition [26], [27], [28] is applied, where we fill in the ghost girds with values extrapolated from the computational boundary to avoid nonphysical reflection at the boundary. The width of the ghost cells is determined by the stencil of the finite difference scheme, and it is set as two in this article to accommodate the ENO2 scheme.

3) Backtracking: After the forward reachable set $R_0[t]$ is obtained for $t\in[0,t_f^*]$, the backtracking algorithm is executed to find the optimal waypoints $\mathbf{P}^*(t_n)\in\partial R_0[t_n]$ from $\mathbf{x}_f$ to $\mathbf{x}_s$. We note that, during backtracking, $\mathbf{x}_f$ should be considered as $[x_f\ y_f\ \theta_f]^T$ with θ_f being the heading angle of the robot at t_f^*. The essential calculation in this process is to obtain the previous waypoint $\mathbf{P}^*(t_{n-1})$ given the current waypoint $\mathbf{P}^*(t_n)$.

One way to achieve the above objective is to apply an ODE-based backtracking algorithm [29], which integrates the kinematic equation(1) one step backward

$$\begin{cases}x(t_{n-1})=x(t_n)-\Delta t\cdot F^*(t_n)\cos(\theta(t_n))\\ y(t_{n-1})=y(t_n)-\Delta t\cdot F^*(t_n)\sin(\theta(t_n))\\ \theta(t_{n-1})=\theta(t_n)-\Delta t\cdot\omega^*(t_n)\end{cases}\tag{21}$$

where $F^*(t_n)$ and $\omega^*(t_n)$ are the optimal controls that can be determined from the maximization operation in (16) (see Appendix A for details) as

$$F^*(t_n)=\begin{cases}F_{\max}, & \text{if } \Theta(t_n)>0\\ 0, & \text{if } \Theta(t_n)\leq 0\end{cases}\tag{22}$$

with

$$\begin{aligned}\Theta(t_n)&=\cos(\theta(t_n))\cdot\phi_x(\mathbf{P}^*(t_n),t_n)\\ &\quad+\sin(\theta(t_n))\cdot\phi_y(\mathbf{P}^*(t_n),t_n)+\frac{|\phi_\theta(\mathbf{P}^*(t_n),t_n)|}{R_{\min}}\end{aligned}\tag{23}$$

and

$$\omega^*(t_n) = \begin{cases} -\dfrac{F^*(t_n)}{R_{\min}}, & \text{if } \phi_\theta(\mathbf{P}^*(t_n), t_n) < 0 \\ 0, & \text{if } \phi_\theta(\mathbf{P}^*(t_n), t_n) = 0 \\ +\dfrac{F^*(t_n)}{R_{\min}}, & \text{if } \phi_\theta(\mathbf{P}^*(t_n), t_n) > 0. \end{cases} \tag{24}$$

Equation (24) implies that the optimal trajectory of the robot contains only straight lines or arcs with minimum turning radius, i.e., a case of bang-bang control.

The ODE-based method for backtracking works for most cases except when 1) there exists a singular point of $\phi(\mathbf{x}, t)$ on the path $\mathbf{P}^*(t)$ (so that (23) and (24) cannot be evaluated) or 2) the path $\mathbf{P}^*(t)$ interacts with the obstacles. The situation 1) can happen, for example, in cases when multiple optimal paths exist, as discussed in detail in [16]. The situation 2) is generally not expected as the optimal path usually only grazes (i.e., touch tangentially) an obstacle at a single point. Therefore, almost everywhere in the state space, the ODE-based method can be considered valid [29]. However, in cases where the optimal path needs to slide on the surface of an obstacle (say with curvature between 0 and $1/R_{\min}$) to reach the destination, it is known [21], [30] that the ODE-based method produces spurious result. This is completely natural because, according to (24), the planned path can only follow the curvature of either 0 or $1/R_{\min}$, not being able to slide on the obstacle surface.

To address this fundamental difficulty in ODE-based backtracking, we propose a new scheme based on the backward reachable set $B_{\mathbf{x}}[-\Delta t]$, which is defined as the set of states that can reach a target state $\mathbf{x}$ with kinematics (1) after a time interval Δt. In principle, given $\mathbf{P}^*(t_n)$, the previous waypoint $\mathbf{P}^*(t_{n-1})$ should lie on both $B_{\mathbf{P}^*(t_n)}[-\Delta t]$ and $\partial R_0[t_{n-1}]$. Therefore, a backtracking algorithm can be developed by searching for the intersections of $B_{\mathbf{P}^*(t_n)}[-\Delta t]$ and $\partial R_0[t_{n-1}]$. Since this method does not rely on the derivative computation in (23) and (24) and can choose $\mathbf{P}^*(t_{n-1})$ following any turning rate ω (within the constraint), it overcomes the difficulties existing in the ODE-based method.

The implementation of the backward-set method involves the computation of $B_{\mathbf{P}^*(t_n)}[-\Delta t]$ for each $\mathbf{P}^*(t_n)$. Fortunately, as $\partial R_0[t]$ already excludes the obstacles thanks to the VI (18), we can ignore the obstacles when computing $B_{\mathbf{P}^*(t_n)}[-\Delta t]$, and $\mathbf{P}^*(t_{n-1})$ selected through the proposed method is still guaranteed to be collision-free. As a consequence, the computation of the backward reachable set only needs to be conducted once for an arbitrary state because the geometries of $B_{\mathbf{P}^*(t_n)}[-\Delta t]$ are the same for each $\mathbf{P}^*(t_n)$. For simplicity, we can precompute $B_0[-\Delta t]$ as the backward set of $\mathbf{x} = [0\ 0\ 0]^T$, which is equivalent to a shifted $R_0[\Delta t]$ emanating from $[0\ 0\ 0]^T$ that can be easily computed by integrating $\mathrm{I} = 0$ in (18).

The geometry of $B_0[-\Delta t]$ computed following this method is a volume in the shape of a ginkgo leaf, as shown in Fig. 5. To determine whether a candidate state $\mathbf{x}_{\text{cand}} \in \partial R_0[t_{n-1}]$ belongs to $B_{\mathbf{P}^*(t_n)}[-\Delta t]$, we can transform $\mathbf{x}_{\text{cand}}$ in global coordinates to $\hat{\mathbf{x}}_{\text{cand}}$ in local coordinates centered at $\mathbf{P}^*(t_n)$ with the local x-axis pointing to $\theta(t_n)$-direction, following:

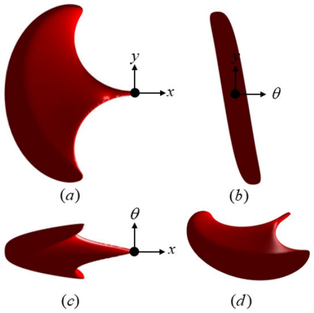


Fig. 5. Geometry of $B_0[-\Delta t]$ from (a) top view, (b) left view, (c) main view, and (d) axonometric view.

$$\begin{bmatrix} \hat{x}_{\text{cand}} \\ \hat{y}_{\text{cand}} \\ \hat{\theta}_{\text{cand}} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta(t_n)) & \sin(\theta(t_n)) & 0 & x(t_n) \\ -\sin(\theta(t_n)) & \cos(\theta(t_n)) & 0 & y(t_n) \\ 0 & 0 & 1 & \theta(t_n) \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} x_{\text{cand}} \\ y_{\text{cand}} \\ \theta_{\text{cand}} \\ 1 \end{bmatrix}. \tag{25}$$

Then, we simply need to judge whether $\hat{\mathbf{x}}_{\text{cand}}$ belongs to $B_0[-\Delta t]$ (based on the sign of the value function to represent $B_0[-\Delta t]$). In summary, our algorithm repeatedly picks candidates $\mathbf{x}_{\text{cand}}$ from $\partial R_0[t_{n-1}]$ until finding a qualified one which also belongs to $B_0[-\Delta t]$ in local coordinates. We remark that in the cases with multiple optimal paths, our algorithm is only required to find one of them, so the above implementation is sufficient.

IV. Receding Horizon Implementation

To apply the algorithm with limited perception capability (so that obstacles of a particular distance away from the robot are not known), we need to utilize the method in a receding horizon. More specifically, we perform the path planning algorithm repeatedly within a finite horizon until the robot reaches the given global target. For simplicity, we consider the finite horizon as a circular area of radius R_H centered at the current position $(x_{\text{cen}}, y_{\text{cen}})$ of the robot, whose boundary is given by

$$\text{Bdry}(x_{\text{cen}}, y_{\text{cen}}) \equiv \left\{(x, y)\ |(x - x_{\text{cen}})^2 + (y - y_{\text{cen}})^2 = R_H^2\right\}. \tag{26}$$

In each finite horizon not enclosing (x_f, y_f), we choose a position $(x_{\text{fi}}, y_{\text{fi}})$ from the boundary of the finite horizon $\text{Bdry}(x_{\text{cen}}, y_{\text{cen}})$ to construct the intermediate target state $\mathbf{x}_{\text{fi}} = [x_{\text{fi}}\ y_{\text{fi}}\ \theta_{\text{fi}}]^T$ with $\theta_{\text{fi}} \in [0, 2\pi]$. An appropriate choice of $\mathbf{x}_{\text{fi}}$ should be in favor of the following two considerations: 1) the trajectory $\mathbf{P}^*(t)$ converges to $\mathbf{x}_f$ and 2) the mobile robot does not go toward the obstacles immediately outside the current finite horizon. Therefore, we select the intermediate target state by $(x_{\text{fi}}, y_{\text{fi}}) = \text{argmin}_{(x,y)}\, c(x, y)$ with $c(x, y)$ a heuristic cost function

$$c(x, y) = d(x, y) + a \times h(x, y) \tag{27}$$

where $d(x, y)$ measures the distance from (x, y) to (x_f, y_f), helping $\mathbf{P}^*(t)$ converge to $\mathbf{x}_f$. a is a weighting factor. $h(x, y)$ evaluates the potential risks of navigating the mobile robot toward (x, y), helping it to avoid potential collisions outside the current horizon. The choices of a and $h(x, y)$ depend on specific applications that will be elaborated in Section V-D.

The full algorithm implemented in a receding horizon together with the backward-set-based backtracking is provided in Algorithm 1. In implementing the algorithm, users should input the initial state $\mathbf{x}_s$, target state $\mathbf{x}_f$, maximal speed $F_{\max}$, minimal turning radius $R_{\min}$ of the robot, and the radius of receding horizon (if the receding horizon is employed). Moreover, users can change the setting of grid spacing and time step size in solving the level set equation based on a balance between the accuracy and computational cost.

Next, we briefly discuss the asymptotic computational complexity of algorithm 1, which mainly comes from evolving (18) in time via the level set method. The standard level set method that solves (18) in the full domain yields a computational complexity of $O(n^3)$ in each time step, with n being the number of grids in each dimension. According to the C-F-L condition, the time step Δt is inversely proportional to n, so that the number of time steps for fixed t_f^* is proportional to n. As a result, the asymptotic computational complexity of algorithm 1 for the full-domain level set method can be estimated by $O(n^4)$. One way to reduce the computational complexity is to resort to a narrowband level set method [13] which solves (18) only within a narrow shell (with bandwidth d) enclosing the reachable front, leading to $O(n^3 d)$ instead of $O(n^4)$ for a solution.

Applications of Algorithm 1 certainly include situations of off-line planning or producing benchmark exact solution for other reduced-order planning methods. For real-time online applications, it is likely that other acceleration method needs to be combined with Algorithm 1. One possibility is parallel implementation of Algorithm 1 on GPU, following previous GPU solution of other level-set equations [31], where the full computational domain is split into subdomains, and each of them is assigned a thread of GPU to work on. Alternatively, one may accelerate Algorithm 1 by applying dimensionality decomposition on the state space to decompose the dynamics of the system into subsystems coupled through common states as in [32]. As we focus primarily on the development and verification of Algorithm 1, these acceleration methods will not be further explored in the scope of current article.

V. Case Study

In this section, we test the proposed time-optimal path planner in four cases. In the first case, we consider a synthetic problem with one moving and deforming obstacle and a known optimal path. The purpose of it is to demonstrate the validity of the proposed algorithm to plan the optimal path. In the second case, we consider the path planning of an unmanned surface vessel (USV) in modeled ocean waves, with multiple evolving dangerous high waves deemed as moving and deforming obstacles. In the third case, we demonstrate the advantage of the proposed backward-set-based backtracking scheme compared to the traditional ODE-based method. Finally, the fourth case is devoted to the receding-horizon implementation of the algorithm in the same ocean wave field as in the second case. In all cases of this section, we set the maximal speed of the vessel $F_{\max} = 4$ m/s, and the minimal turning radius $R_{\min} = 4$ m. The target heading θ_f can be an arbitrary number within $[0, 2\pi]$.

Algorithm 1 Path Planning in a Receding Horizon

```
INPUT: Radius $R_H$, Target state $\mathbf{x}_f$, Maximal speed $F_{\max}$
       Initial state $\mathbf{x}_s$, Minimal turning radius $R_{\min}$
OUTPUT: Optimal waypoints set W for each horizon
▷ Determine the center of current horizon
$\mathbf{x}_{\text{cen}} \leftarrow \mathbf{x}_s$
while $|\mathbf{x}_{\text{cen}} - \mathbf{x}_f| > \epsilon$
  ▷ Selection of Intermediate Target
  if $(x_f - x_{\text{cen}})^2 + (y_f - y_{\text{cen}})^2 \le R_H^2$
    $\mathbf{x}_{\text{fi}} \leftarrow \mathbf{x}_f$
  else
    $\mathbf{x}_{\text{fi}} \leftarrow \{\mathbf{x} \in R^3 \,|(x, y) = \text{argmin } c(x, y)\}$
  end
  ▷ Forward propagation
  while $\phi(\mathbf{x}_{\text{fi}}, t_n) > 0$
    prediction with (19)
    correction with (20)
    n ← n+1
  end
  ▷ Backtracking
  $\mathbf{P}^*(t_n) \leftarrow \mathbf{x}_{\text{fi}}$        ▷ current waypoint $\mathbf{P}^*(t_n)$
  W ← $\mathbf{x}_{\text{fi}}$
  while $|\mathbf{P}^*(t_n) - \mathbf{x}_{\text{cen}}| > \epsilon$
    for each $\mathbf{x}_{\text{cand}} \in \{\mathbf{x} \in \mathbb{R}^3 | \phi(\mathbf{x}, t_{\text{n-1}}) = 0\}$
      $\hat{\mathbf{x}}_{\text{cand}} \leftarrow$ coordinate transformation of $\mathbf{x}_{\text{cand}}$ via (25)
      if $\hat{\mathbf{x}}_{\text{cand}} \in B_0[-\Delta t]$
        $\mathbf{P}^*(t_{\text{n-1}}) \leftarrow \mathbf{x}_{\text{cand}}$
        W ← W $\bigcup \mathbf{P}^*(t_{\text{n-1}})$
        n ← n-1
        break
      end
    end
  end
  return W
  $\mathbf{x}_{\text{cen}} \leftarrow \mathbf{x}_{\text{fi}}$;
end
```

We remark that, to strike a balance between academic rigor and practical relevance, all tests of USV will be centered on the kinematic model (1), which is a simplification relative to realistic cases. For actual applications of path planning in USV, one may need to further consider a maneuvering model such as marine maneuvering game (MMG) model [33] with the thrust and rudder angle being controlled. It may also be beneficial to include USV dynamics in waves through a vessel seakeeping model [34], to be implemented in conjunction with (1) as well as more constraints regarding the vessel motion. These considerations of the control architecture will introduce additional complexities in implementing the planning algorithm, which we intend to address in the future.

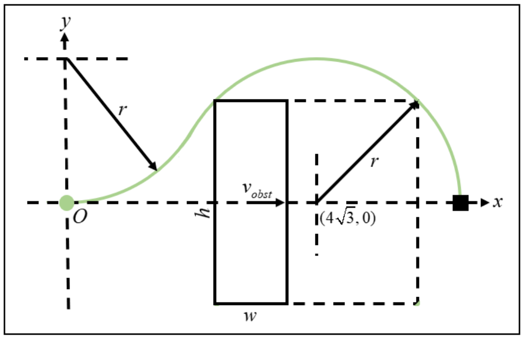


Fig. 6. Problem setup for the case in Section V-A, including the projections of $\mathbf{x}_s$(●), $\mathbf{x}_f$(■), and $\mathbf{P}^*(t)$(——) on the xy plane, the obstacle (——), and the envelope of the obstacle's trajectory (– – –).

A. Synthetic Case With a Single Obstacle

The problem setup of the synthetic case is shown in Fig. 6 with $\mathbf{x}_s = [0\ 0\ 0]^T$ and $\mathbf{x}_f = [4 + 4\sqrt{3}\ 0\ \theta_f]^T$. We consider a moving and deforming rectangular obstacle with a height of $h = 4\sqrt{2}m$ and a width $w(t)$ varying in accordance with

$$w(t) = \begin{cases} 2, & \text{if } t \notin \left[\frac{5\pi}{12}, \frac{11\pi}{12}\right] \\ 2 + 2 \cdot \sin\left(2t - \frac{5\pi}{6}\right), & \text{if } t \in \left[\frac{5\pi}{12}, \frac{11\pi}{12}\right]. \end{cases} \tag{28}$$

The center of the obstacle is located at $(1 + 4\sqrt{3} - 2\sqrt{2}, 0)$ at $t = 0$, and the center travels along the positive x-direction with

$$v_{\text{obst}}(t) = \begin{cases} 0, & \text{if } t \notin \left[\frac{5\pi}{12}, \frac{11\pi}{12}\right] \\ \frac{8\sqrt{2} - 4}{\pi} \text{ m/s}, & \text{if } t \in \left[\frac{5\pi}{12}, \frac{11\pi}{12}\right]. \end{cases} \tag{29}$$

The area occupied by the full trajectory of the obstacle is a square inscribed to a circle centered at $(4\sqrt{3}, 0)$ with radius $r = 4$ m. Therefore, the exact optimal path (on the xy plane) comprises of two circular arcs with $R_{\min}$ as shown in Fig. 6, corresponding to an exact optimal arrival time 3.6652 s. Fig. 7 plots the results of our algorithm under the grid size $\Delta x = \Delta y = 7/60$ m and $\Delta\theta = \pi/120$ rad, where the robot indeed follows a curve containing two circular arcs and arrives at $\mathbf{x}_f$ with $t_f^* = 3.6946$ s. The small deviation of t_f^* from the exact result is mainly caused by the fact that the initial state $\mathbf{x}_s$ needs to be inflated to be $\mathbf{X}_s$ (as discussed in [26] and Section III-A) to maintain the robustness of the numerical computation. For this case, we use $\mathbf{X}_s$ as a small ball with radius of two-grid size.

B. Case of USV in Ocean Waves

In this example, an USV is navigating in an ocean wave field that is described by a JONSWAP spectrum [35] with significant wave height $H_s = 5$ m (sea state 6) and spreading angle $\theta = 30°$. The wave field is generated by a nonlinear wave model—the high-order spectral (HOS) method [36], [37], [38], [39]. The computational workspace of the USV is set as a rectangular area of size 50 m × 20 m, with grid sizes $\Delta x = \Delta y = 0.2$ m and $\Delta\theta = \pi/50$ rad. The USV travels from $\mathbf{x}_s = [44\ 10\ 0.75\pi]^T$ to $\mathbf{x}_f = [4\ 10\ \theta_f]^T$. The extreme waves (any regions with surface elevation beyond ±3 m) are treated as moving and deforming obstacles which should be avoided by the USV. The results of our path planning algorithm are shown in Fig. 8 with $\hat{t}_f^* = 13.9531$ s. We see that the path planner provides an optimal path $\mathbf{P}^*(t)$ that avoids all dangerous waves, which is also confirmed to satisfy the kinematic constraint of $R_{\min}$.

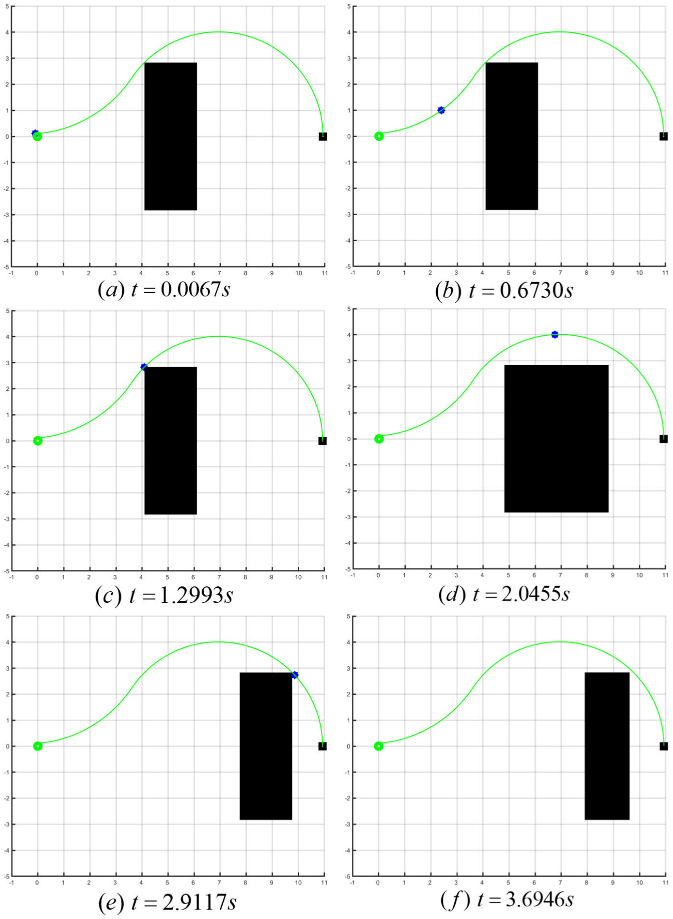
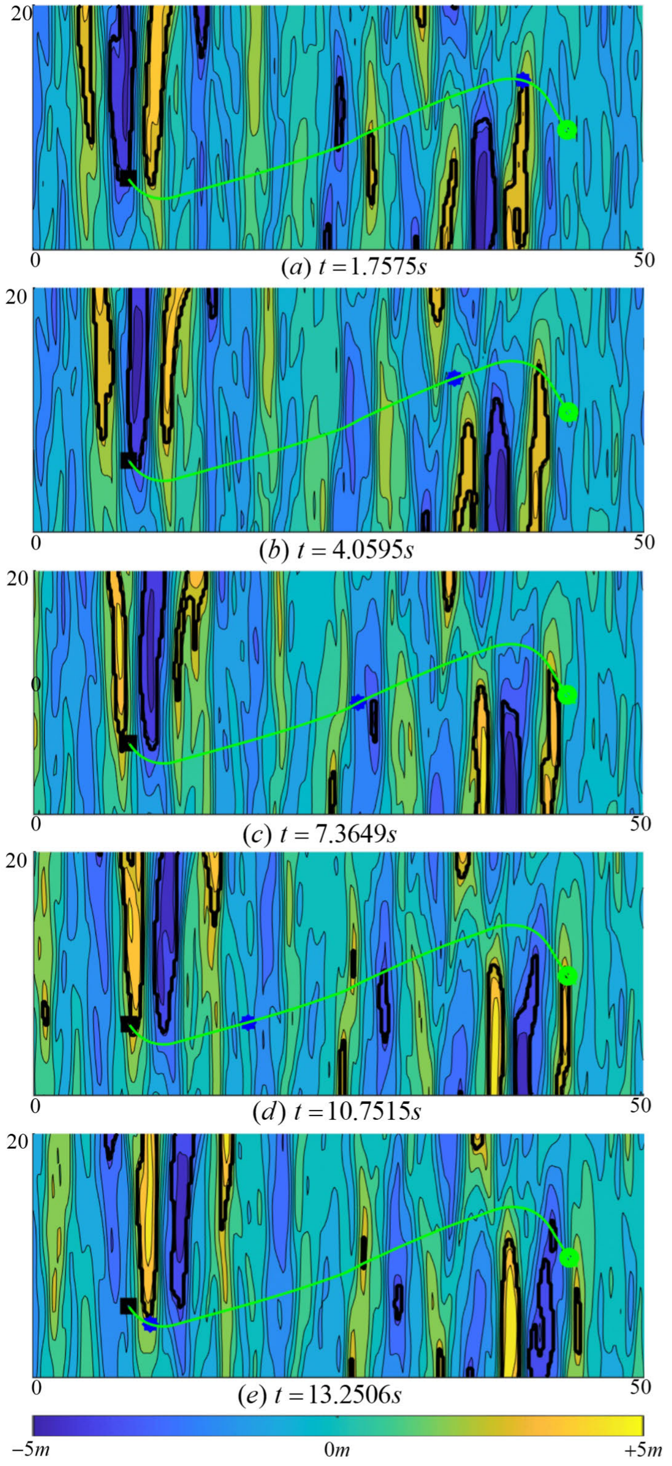


Fig. 7. Snapshots of the solution for the problem in Section V-A with incremental change of $t \in [0, t_f^*]$ in (a)–(f), including the obstacle (black shadow), the projections of $\mathbf{P}^*(t)$(——), $\mathbf{x}_s$(●), $\mathbf{x}_f$(■), and $\mathbf{P}^*(t_n)$(∗) on the xy plane. (a) $t = 0.0067$ s. (b) $t = 0.6730$ s. (c) $t = 1.2993$ s. (d) $t = 2.0455$ s. (e) $t = 2.9117$ s. (f) $t = 3.6946$ s.

C. Case With Different Backtracking Algorithms

We next demonstrate the advantage of the backward-set-based algorithm over the ODE-based algorithm for backtracking. For this purpose, we design a simple test case involving one static circular obstacle of a radius 5 m (i.e., the curvature of its boundary falls into $(0, 1/R_{\min})$). The initial and target state are set to be $\mathbf{x}_s = [-5\ 0\ 0.5\pi]^T$ and $\mathbf{x}_f = [0\ 5\ \theta_f]^T$, respectively. As a result, the time-optimal path should be a circular arc tracing the boundary of the obstacle.

As mentioned in Section III-B, the ODE-based backtracking scheme cannot exactly trace the obstacle boundary since the planned path can only follow curves with radius $R_{\min}$. This is shown in Fig. 9(a) where we see a clear deviation of the



(a) $t = 1.7575s$

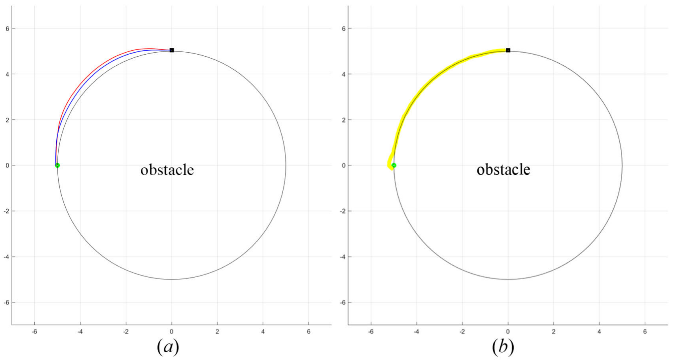
(b) $t = 4.0595s$

(c) $t = 7.3649s$

(d) $t = 10.7515s$

(e) $t = 13.2506s$

Fig. 8. Snapshots of the solution for the problem with realistic ocean waves in Section V-B with incremental change of $t \in [0, \hat{t}_f^*]$ in (a)–(e), including the projections of $A_{\text{obst}}(t)$ (enclosed by ——), $\mathbf{x}_s$ (●), $\mathbf{x}_f$ (■), $\mathbf{P}^*(t)$ (——), and $\mathbf{P}^*(t_n)$ ($*$) on the xy plane. The instantaneous surface elevation fields are denoted by color contours in all subfigures. (a) $t = 1.7575$ s. (b) $t = 4.0595$ s. (c) $t = 7.3649$ s. (d) $t = 10.7515$ s. (e) $t = 13.2506$ s.



(a)

(b)

Fig. 9. Optimal paths $\mathbf{P}^*(t)$ projected on the xy plane, planned by (a) ODE-based method with $\Delta t = 7/30$ s (——), and $\Delta t = 7/60$ s(——) and (b) backward-set-based method with $\Delta t = 7/30$ s (——). The projection of starting and target points is shown with $\mathbf{x}_s$ (●) and $\mathbf{x}_f$ (■).

planned path from the obstacle boundary. While decreasing the time step size Δt [from 7/30 to 7/60 s as shown in Fig. 9(a)] can alleviate the error, the convergence to the exact optimal path is in general very slow. On the other hand, since the backward-set-based method relies only on the geometric relation between $\partial R_0[t_{n-1}]$ and $B_0[-\Delta t]$, the planned path can follow any desired curvature. As demonstrated in Fig. 9(b), the backward-set-based scheme provides a very accurate optimal path even with $\Delta t = 7/30$ s.

D. Case With Implementation in Receding Horizons

Finally, we demonstrate the receding horizon implementation of the path planner for an USV in the same ocean wave field as considered in Section V-B. We assume a finite horizon of $R_H = 10$ m. To compute the heuristic cost function $c(x, y)$ in (27), we consider $h(x, y)$ as the standard deviation of surface elevation in an interval of five wave periods, starting from the time when the planning process begins (a larger standard deviation implies more violent fluctuation of the surface). A weighting factor of $a = 2$ is applied (based on trial-and-errors).

The planned optimal path is shown in Fig. 10. We see that the USV successfully avoids all obstacles and reaches $\mathbf{x}_f$ at $t_f^* = 14.7547$ s. Clearly this is a suboptimal solution compared to the global optimal solution realized in Section V-B with $t_f^* = 13.9531$ s. The suboptimality is not avoidable due to the limited perception horizon R_H in applications but may be further improved by designing more effective heuristic cost function $c(\mathbf{x})$ to select intermediate target states with the heading angle θ taken into consideration.

VI. Conclusion

In this article, we develop a time-optimal path planning algorithm for a mobile robot constrained by its minimum turning radius and maximum traveling speed, in an environment cluttered with an arbitrary number of moving and deforming obstacles. The algorithm is based on the reachability theory implemented in a 3-D state space $[x\ y\ \theta]^T$ of the robot. The forward propagation of the reachable set is solved through a VI (which we prove), where only the location of the obstacles at each time instant is required. The backtracking for waypoints on the optimal path is solved through an ODE-based method, or a more robust newly developed scheme based on the backward reachable set. The full path planner is tested in four cases to demonstrate its multiple merits, including applications of USV to an ocean wave field.

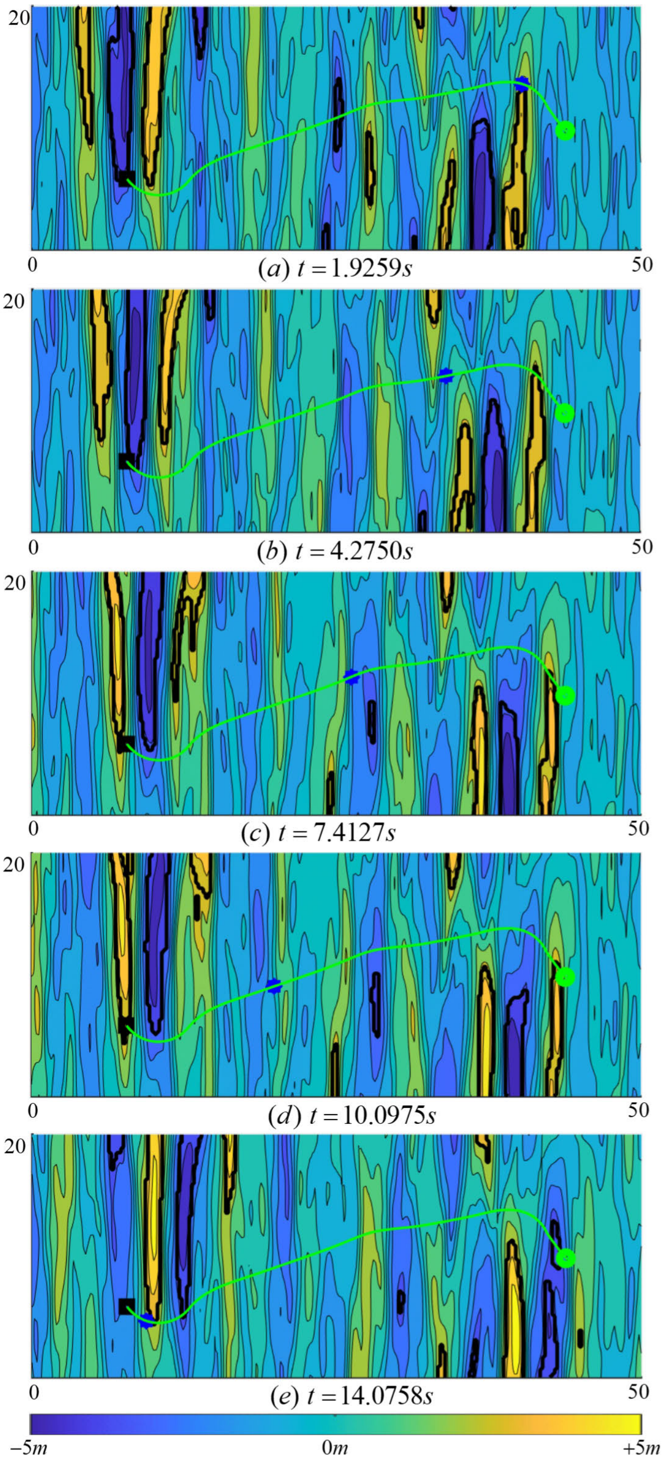


Fig. 10. Snapshots of the solution for the problem with realistic ocean waves in Section V-D with incremental change of $t \in [0, t_f^*]$ in (a)–(e), including the projections of $A_{\text{obst}}(t)$ (enclosed by ——), $\mathbf{x}_s$ (●), $\mathbf{x}_f$ (■), $\mathbf{P}^*(t)$ (——), and $\mathbf{P}^*(t_n)$ ($*$) on xy plane. The instantaneous surface elevation fields are denoted by color contours in all subfigures. (a) $t = 1.9259$ s. (b) $t = 4.2750$ s. (c) $t = 7.4127$ s. (d) $t = 10.0975$ s. (e) $t = 14.0758$ s.

Finally, we comment that uncertainties in the measurement and prediction of positions of obstacles would impose another dimension in consideration of the path planning. A potential way to take such uncertainty into account is to switch to a probabilistic framework. More specifically, one may perform formal safety analysis via probabilistic reachability computation and rule out states from the reachable set whose potential risk of collision is beyond a predetermined threshold. With more developments in this direction, the algorithm can consider the probabilistic risk and arrive at a path whose collision risk is under a controlled level.

Appendix A
Derivation of Equation (17) and Optimal Control

In this appendix, we detail: 1) the derivation for (17) and 2) the selection of optimal controls in (22) and (24). To realize 1), we start from (15) and perform the sup operator by choosing an optimal $\omega^*(t)$ for any $F(t) \in [0, F_{\max}]$. As $F(t) \geq 0$ is independent of $\omega(t)$, for arbitrary $F(t)$, $\omega^*(t)$ can be selected as

$$\omega^*(t) = \begin{cases} -\dfrac{F(t)}{R_{\min}}, & \text{if } \phi_\theta(\mathbf{x}, t) < 0 \\ \text{arbitrary}, & \text{if } \phi_\theta(\mathbf{x}, t) = 0 \\ +\dfrac{F(t)}{R_{\min}}, & \text{if } \phi_\theta(\mathbf{x}, t) > 0. \end{cases} \tag{30}$$

In this case, (15) is reduced to

$$\phi_t + \sup_{F(t)}\{F(t) \cdot \Theta(\mathbf{x}, t)\} = 0 \tag{31}$$

with $\Theta(\mathbf{x}, t) = \cos\theta \cdot \phi_x(\mathbf{x}, t) + \sin\theta \cdot \phi_y(\mathbf{x}, t) + |\phi_\theta(\mathbf{x}, t)|/R_{\min}$. We then select $F^*(t)$ based on (31), which leads to

$$F^*(t) = \begin{cases} F_{\max}, & \text{if } \Theta(t_n) > 0 \\ 0, & \text{if } \Theta(t_n) \leq 0. \end{cases} \tag{32}$$

Substituting (30) and (32) into (15), we arrive at (17). Next, we present a detailed proof by contradiction to verify the correctness of the above process.

Proof: Suppose that $\omega^*(t)$ and $F^*(t)$ are not optimal, and let $F^o(t)$ and $\omega^o(t)$ be the optimal controls for

$$\sup_{\omega(t), F(t) \in \mathcal{U}} \{F(t)\phi_x \cos\theta + F(t)\phi_y \sin\theta + \phi_\theta\ \omega(t)\}. \tag{33}$$

It is easy to see that

$$\begin{aligned} &\{F^o(t)\phi_x \cos\theta + F^o(t)\phi_y \sin\theta + \phi_\theta\ \omega^o(t)\} \\ &\quad \geq \{F^*(t)\phi_x \cos\theta + F^*(t)\phi_y \sin\theta + \phi_\theta\ \omega^*(t)\}. \end{aligned} \tag{34}$$

Since $\forall F(t) \in [0, F_{\max}]$, we have

$$\begin{aligned} &\sup_{\omega(t)}\{F(t)\phi_x \cos\theta + F(t)\phi_y \sin\theta + \phi_\theta\ \omega(t)\} \\ &\quad = \{F(t)\phi_x \cos\theta + F(t)\phi_y \sin\theta + \phi_\theta\ \omega^*(t)\} \\ &\quad = \left\{F(t)\phi_x \cos\theta + F(t)\phi_y \sin\theta + |\phi_\theta| \frac{F(t)}{R_{\min}}\right\} \\ &\quad \geq \left\{F(t)\phi_x \cos\theta + F(t)\phi_y \sin\theta + \phi_\theta\ \omega^o(t)\right\}. \end{aligned} \tag{35}$$

As $F(t)$ is independent of $\omega(t)$, we can obtain

$$\begin{aligned} &\{F^*(t)\phi_x \cos\theta + F^*(t)\phi_y \sin\theta + \phi_\theta\ \omega^*(t)\} \\ &\quad = \sup_{F(t)}\left\{F(t)\phi_x \cos\theta + F(t)\phi_y \sin\theta + |\phi_\theta| \frac{F(t)}{R_{\min}}\right\} \\ &\quad \geq \left\{F^o(t)\phi_x \cos\theta + F^o(t)\phi_y \sin\theta + |\phi_\theta| \frac{F^o(t)}{R_{\min}}\right\} \\ &\quad \geq \{F^o(t)\phi_x \cos\theta + F^o(t)\phi_y \sin\theta + \phi_\theta\ \omega^o(t)\}. \end{aligned} \tag{36}$$

Combining (34) and (36), we arrive at

$$\begin{aligned}&\{F^o(t)\phi_x\cos\theta + F^o(t)\phi_y\sin\theta + \phi_\theta\ \omega^o(t)\}\\ &\quad = \{F^*(t)\phi_x\cos\theta + F^*(t)\phi_y\sin\theta + \phi_\theta\ \omega^*(t)\}\end{aligned} \tag{37}$$

which leads to a contradiction. As a result, $\omega^*(t)$ and $F^*(t)$ are the optimal controls. ■

For 2), (22) and (24) can be understood from (30) and (32) except that more treatments are needed for the case of $\phi_\theta(\mathbf{x}, t) = 0$ in (30). While arbitrary choices of $\omega^*(t)$ is allowed for the forward propagation of $\phi(\mathbf{x}, t)$ via the VI, we can show that on the optimal trajectory $\mathbf{P}^*(t)$, $\omega^*(t) = 0$ if we have $\phi_\theta(\mathbf{x}, t) = 0$ as follows.

We start by rewriting (15) in vector form as

$$\phi_t + \nabla\phi \cdot Q(\mathbf{x}, t) = 0 \tag{38}$$

where $Q(\mathbf{x}, t) = [F^*(t)\cos\theta\ \ F^*(t)\sin\theta\ \ \omega^*(t)]^T$. Taking the partial derivative with respect to $\mathbf{x}$ on both sides of (38), we obtain

$$\nabla(\phi_t) + \nabla(\nabla\phi) \cdot Q + \nabla Q \cdot \nabla\phi = \mathbf{0}. \tag{39}$$

We now formulate (39) on the optimal path $\mathbf{P}^*(t)$ so that $Q(\mathbf{P}^*(t), t) = \dot{\mathbf{P}}^*(t)$, and the first two terms in (39) can be written as the total derivative $d(\nabla\phi)/dt$ following the path $\mathbf{P}^*(t)$. Therefore, (39) becomes

$$\frac{d}{dt}(\nabla\phi) = \begin{bmatrix} d\phi_x/dt \\ d\phi_y/dt \\ d\phi_\theta/dt \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ F^*(t)\big(\sin\theta\phi_x - \cos\theta\phi_y\big) \end{bmatrix}. \tag{40}$$

Since $\phi_\theta = 0$ on $\mathbf{P}^*(t)$, we have $F^*(t)(\sin\theta\ \phi_x - \cos\theta\ \phi_y) = 0$ from the third equation in (40). For $F^*(t) = F_{\max}$, ϕ_x and ϕ_y have to be nonzero constants (from the first two equations in (40), and thus, θ has to stay a constant, i.e., $\omega^*(t) = 0$. For $F^*(t) = 0$, $\omega^*(t)$ also has to be zero due to the definition $\omega(t) \in [-F(t)/R_{\min}, F(t)/R_{\min}]$.

Appendix B
Proof of Variational Inequality (18)

In this section, we prove that $\phi(\mathbf{x}, t)$ is the unique continuous viscosity solution of (18) with initial condition (13). To do this, we first define what we mean by viscosity solution of (18).

Definition 1 (Viscosity Solution [40], [41]): An upper semicontinuous (resp., lower semicontinuous) function $\phi(\mathbf{x}, t)$ is a viscosity subsolution (resp., supersolution) of (18) if for any $(\mathbf{x}, t)$ and any test function $\Psi(\mathbf{x}, t) \in C^1$ such that $\Psi = \phi$ at point $(\mathbf{x}, t)$, and $\phi - \Psi \le 0$ (resp., $\phi - \Psi \ge 0$) in a neighborhood of $(\mathbf{x}, t)$, then Ψ has to satisfy

$$\begin{aligned}&\min\{\Psi_t + H(\mathbf{x}, \nabla\Psi), \Psi(\mathbf{x}, t) - g(\mathbf{x}, t)\} \le 0\\ &(\text{resp., } \min\{\Psi_t + H(\mathbf{x}, \nabla\Psi), \Psi(\mathbf{x}, t) - g(\mathbf{x}, t)\} \ge 0)\end{aligned} \tag{41}$$

where the Hamiltonian $H(\mathbf{x}, \nabla\Psi)$ is

$$\begin{aligned}H &= \sup_{\omega(t), F(t) \in \mathcal{U}} \{\nabla\Psi \cdot \dot{\mathbf{P}}(t)\}\\ &= \max\left(0,\ F_{\max}\Psi_x\cos\theta + F_{\max}\Psi_y\sin\theta + |\Psi_\theta|\frac{F_{\max}}{R_{\min}}\right).\end{aligned} \tag{42}$$

$\phi(\mathbf{x}, t)$ is a viscosity solution of (18) if it is a viscosity subsolution and a viscosity supersolution of (18).

Before carrying on the proof, we review the application of the dynamic programming principle [42] to our definition of $\phi(\mathbf{x}, t)$ [as in (12)] in the following lemma.

Lemma 1 (Dynamic Programming Principle): For $\Delta t \ge 0$, the value function $\phi(\mathbf{x}, t)$ defined by (12) has to satisfy

$$\begin{aligned}\phi(\mathbf{x}, t) = \operatorname*{Inf}_{\omega(h), F(h) \in \mathcal{U}} \Big\{&\max\big(\phi(\mathbf{y}_{\mathbf{x}}^{\omega,F}(t - \Delta t), t - \Delta t),\\ &\max_h g(\mathbf{y}_{\mathbf{x}}^{\omega,F}(h), h)\big)\Big\}\end{aligned} \tag{43}$$

where $h \in [t - \Delta t, t]$.

Lemma 1 can be proved by contradiction, and interested readers can refer to [42] and references therein for details.

In the following part, we verify that $\phi(\mathbf{x}, t)$ as defined in (12) is the viscosity solution to (18) by checking its supersolution and subsolution properties explained in the definition 1.

Proof: Part 1: We first examine the supersolution property of $\phi(\mathbf{x}, t)$. Let $\Psi(\mathbf{x}, t) \in C^1$ be a test function such that $\phi = \Psi$ at $(\mathbf{x}, t)$, and $\phi - \Psi \ge 0$ in the vicinity of $(\mathbf{x}, t)$. Since $\phi - \Psi \ge 0$ near $(\mathbf{x}, t)$, we have

$$\begin{aligned}&\operatorname*{Inf}_{\omega, F \in \mathcal{U}} \{\phi(\mathbf{y}_{\mathbf{x}}^{\omega,F}(t - \Delta t), t - \Delta t)\}\\ &\quad \ge \Psi(\mathbf{y}_{\mathbf{x}}^{\omega^*,F^*}(t - \Delta t), t - \Delta t)\\ &\quad \ge \operatorname*{Inf}_{\omega, F \in \mathcal{U}} \{\Psi(\mathbf{y}_{\mathbf{x}}^{\omega,F}(t - \Delta t), t - \Delta t)\}\end{aligned} \tag{44}$$

where $\omega^*(\tau)$, $F^*(\tau)$, $\tau \in [t - \Delta t, t]$ denote the optimal controls to minimize $\phi(\mathbf{y}_{\mathbf{x}}^{\omega,F}(t - \Delta t), t - \Delta t)$. It follows that:

$$\begin{aligned}&\Psi(\mathbf{x}, t) - \operatorname*{Inf}_{\omega(h), F(h) \in \mathcal{U}} \{\Psi(\mathbf{y}_{\mathbf{x}}^{\omega,F}(t - \Delta t), t - \Delta t)\}\\ &\quad \ge \phi(\mathbf{x}, t) - \operatorname*{Inf}_{\omega(h), F(h) \in \mathcal{U}} \{\phi(\mathbf{y}_{\mathbf{x}}^{\omega,F}(t - \Delta t), t - \Delta t)\} \ge 0\end{aligned} \tag{45}$$

where the second inequality follows from (43).

Expanding $\Psi(\mathbf{y}_{\mathbf{x}}^{\omega,F}(t - \Delta t), t - \Delta t)$ at $(\mathbf{x}, t)$ via the Taylor series, we obtain

$$\begin{aligned}&\Psi(\mathbf{y}_{\mathbf{x}}^{\omega,F}(t - \Delta t), t - \Delta t)\\ &\quad = \Psi(\mathbf{x}, t) + \nabla\Psi \cdot (\mathbf{y}_{\mathbf{x}}^{\omega,F}(t - \Delta t) - \mathbf{x}) - \Psi_t \cdot \Delta t\\ &\qquad + o(\mathbf{y}_{\mathbf{x}}^{\omega,F}(h) - \mathbf{x}) + o(\Delta t).\end{aligned} \tag{46}$$

Based on the kinematic relation (1), we have

$$\mathbf{y}_{\mathbf{x}}^{\omega,F}(t - \Delta t) - \mathbf{x} = -\dot{\mathbf{P}}(t) \cdot \Delta t + o(\Delta t). \tag{47}$$

After substituting (46) and (47) into (45), we obtain

$$\begin{aligned}&\Psi_t \cdot \Delta t + \sup_{\omega(t), F(t) \in \mathcal{U}} \{\nabla\Psi \cdot \dot{\mathbf{P}}(t) \cdot \Delta t\} + o(\Delta t)\\ &\quad = \Psi_t \cdot \Delta t + H(\mathbf{x}, \nabla\Psi) \cdot \Delta t + o(\Delta t) \ge 0.\end{aligned} \tag{48}$$

With $\Delta t \to 0$, (48) reduces to

$$\Psi_t + H(\mathbf{x}, \nabla\Psi) \ge 0. \tag{49}$$

According to (12), at $(\mathbf{x}, t)$, we have

$$\Psi(\mathbf{x}, t) = \phi(\mathbf{x}, t) \ge \operatorname*{Inf}_{\omega(\tau), F(\tau) \in \mathcal{U}} \{\max_{\tau \in [0,t]} g(\mathbf{y}_{\mathbf{x}}^{\omega,F}(\tau), \tau)\} \ge g(\mathbf{x}, t). \tag{50}$$

Combining (49) and (50), we have

$$\min\{\Psi_t + H(\mathbf{x}, \nabla\Psi), \Psi(\mathbf{x}, t) - g(\mathbf{x}, t)\} \geq 0. \tag{51}$$

Hence, $\phi(\mathbf{x}, t)$ is a supersolution of (18).

Part 2: We next check the subsolution property of $\phi(\mathbf{x}, t)$. Let $\Psi(\mathbf{x}, t) \in C^1$ be a test function such that $\phi = \Psi$ at $(\mathbf{x}, t)$, and $\phi - \Psi \leq 0$ in the vicinity of $(\mathbf{x}, t)$.

Part 2.1: If $\Psi(\mathbf{x}, t) = \phi(\mathbf{x}, t) \leq g(\mathbf{x}, t)$, then $\Psi(\mathbf{x}, t) - g(\mathbf{x}, t) \leq 0$, and (41) holds trivially.

Part 2.2: If $\Psi(\mathbf{x}, t) = \phi(\mathbf{x}, t) > g(\mathbf{x}, t)$, then based on the continuity of $g(\mathbf{x}, t)$ and $\phi(\mathbf{x}, t)$, there exists a sufficiently small Δt such that $\mathbf{y}_{\mathbf{x}}^{\omega,F}(h)$, $h \in [t - \Delta t, t]$ stays outside the areas occupied by $g(\mathbf{x}, t) > 0$ [17], [44]. As a result, according to (43), we have

$$\phi(\mathbf{x}, t) = \operatorname*{Inf}_{\omega(h), F(h) \in \mathcal{U}} \left\{\phi\left(\mathbf{y}_{\mathbf{x}}^{\omega,F}(t - \Delta t), t - \Delta t\right)\right\}. \tag{52}$$

Similarly, we have

$$\begin{aligned} &\operatorname*{Inf}_{\omega, F \in \mathcal{U}} \{\Psi(\mathbf{y}_{\mathbf{x}}^{\omega,F}(t - \Delta t), t - \Delta t)\} \\ &\quad \geq \phi(\mathbf{y}_{\mathbf{x}}^{\omega^*,F^*}(t - \Delta t), t - \Delta t) \\ &\quad \geq \operatorname*{Inf}_{\omega, F \in \mathcal{U}} \{\phi(\mathbf{y}_{\mathbf{x}}^{\omega,F}(t - \Delta t), t - \Delta t)\} \end{aligned} \tag{53}$$

where $\omega^*(\tau)$, $F^*(\tau)$, $\tau \in [t - \Delta t, t]$ denote the optimal controls to minimize $\Psi(\mathbf{y}_{\mathbf{x}}^{\omega,F}(t - \Delta t), t - \Delta t)$. Therefore

$$\begin{aligned} &\Psi(\mathbf{x}, t) - \operatorname*{Inf}_{\omega, F \in \mathcal{U}} \{\Psi(\mathbf{y}_{\mathbf{x}}^{\omega,F}(t - \Delta t), t - \Delta t)\} \\ &\quad \leq \phi(\mathbf{x}, t) - \operatorname*{Inf}_{\omega, F \in \mathcal{U}} \{\phi(\mathbf{y}_{\mathbf{x}}^{\omega,F}(t - \Delta t), t - \Delta t)\} = 0 \end{aligned} \tag{54}$$

where the equality follows from (52). Via a similar Taylor expansion of $\Psi(\mathbf{y}_{\mathbf{x}}^{\omega,F}(t - \Delta t), t - \Delta t)$ at $(\mathbf{x}, t)$ as in (46) and (47), we get

$$\Psi_t + H(\mathbf{x}, \nabla\Psi) \leq 0 \tag{55}$$

and as a result, we obtain

$$\min\{\Psi_t + H(\mathbf{x}, \nabla\Psi), \Psi(\mathbf{x}, t) - g(\mathbf{x}, t)\} \leq 0. \tag{56}$$

Hence, $\phi(\mathbf{x}, t)$ is a viscosity subsolution of (18).

To sum up, $\phi(\mathbf{x}, t)$ is both a viscosity subsolution and a supersolution of (18), and as a consequence, it is a continuous viscosity solution of (18). In addition, following the comparison principle [41], $\phi(\mathbf{x}, t)$ is the unique continuous viscosity solution of (18) given the initial condition (13). ■

We remark that this proof builds upon [17] but contains substantial improvements. In the original proof in [17] for general VI, the authors accidentally proved the properties (51) and (56) in terms of the function $\phi(\mathbf{x}, t)$ instead of the test function $\Psi(\mathbf{x}, t)$. The latter is needed for the definition of viscosity solution.

Appendix C
Numerical Scheme

To solve (18), we use a combination of the second-order essentially nonoscillatory (ENO2) scheme and the Godunov scheme to compute the spatial derivative ϕ_x, ϕ_y, and ϕ_θ. In particular, ENO2 estimates the forward difference ϕ_x^+, ϕ_y^+, ϕ_θ^+ and backward difference ϕ_x^-, ϕ_y^-, ϕ_θ^- by (taking x-direction at grid point i as an example)

$$\begin{cases} (\phi_x^-)_i = \Phi^1_{i-\frac{1}{2}} + \mathrm{minmod}\left\{\Phi^2_{i-1}, \Phi^2_i\right\}(x_i - x_{i-1}) \\ (\phi_x^+)_i = \Phi^1_{i+\frac{1}{2}} + \mathrm{minmod}\left\{\Phi^2_i, \Phi^2_{i+1}\right\}(x_i - x_{i+1}) \end{cases} \tag{57}$$

where Φ^1 and Φ^2 are first- and second-order divided differences defined as

$$\Phi^1_{i+\frac{1}{2}} = \frac{\phi_{i+1} - \phi_i}{\Delta x}, \quad \Phi^2_i = \frac{\Phi^1_{i+\frac{1}{2}} - \Phi^1_{i-\frac{1}{2}}}{2\Delta x} \tag{58}$$

with

$$\mathrm{minmod}\{\alpha, \beta\} = \begin{cases} \alpha, & \text{if } \alpha\beta > 0 \text{ and } |\alpha| \leq |\beta| \\ \beta, & \text{if } \alpha\beta > 0 \text{ and } |\alpha| > |\beta| \\ 0, & \text{if } \alpha\beta \leq 0. \end{cases} \tag{59}$$

The Hamiltonian for (18) $H(\mathbf{x}, \nabla\phi)$ is then constructed by the Godunov scheme as

$$\hat{H} = \mathrm{ext}_x \mathrm{ext}_y \mathrm{ext}_\theta H(\mathbf{x}, \nabla\phi) \tag{60}$$

with (taking x-direction as an example)

$$\mathrm{ext}_x H = \begin{cases} \min\limits_{\phi_x \in I^x} H, & \text{if } \phi_x^- < \phi_x^+ \\ \max\limits_{\phi_x \in I^x} H, & \text{if } \phi_x^- \geq \phi_x^+ \end{cases} \tag{61}$$

and $I^x = [\min(\phi_x^-, \phi_x^+), \max(\phi_x^-, \phi_x^+)]$.

Acknowledgment

The authors would like to thank Prof. Guangyao Wang and Prof. Jing Sun for valuable suggestions and comments.

References

[1] L. Lapierre, R. Zapata, and P. Lepinay, "Combined path-following and obstacle avoidance control of a wheeled robot," *Int. J. Robot. Res.*, vol. 26, no. 4, pp. 361–375, Apr. 2007.

[2] J. Latombe, *Robot Motion Planning*. New York, NY, USA: Springer, 1997.

[3] H. Kim, S.-H. Kim, M. Jeon, J. Kim, S. Song, and K.-J. Paik, "A study on path optimization method of an unmanned surface vehicle under environmental loads using genetic algorithm," *Ocean Eng.*, vol. 142, pp. 616–624, Sep. 2017.

[4] I. S. Dolinskaya and A. Maggiar, "Time-optimal trajectories with bounded curvature in anisotropic media," *Int. J. Robot. Res.*, vol. 31, no. 14, pp. 1761–1793, Dec. 2012.

[5] Y. Liu and R. Bucknall, "Path planning algorithm for unmanned surface vehicle formations in a practical maritime environment," *Ocean Eng.*, vol. 97, pp. 126–144, Mar. 2015.

[6] J. J. Kuffner and S. M. La Valle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2, Apr. 2000, pp. 995–1001.

[7] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous Robot Vehicles*. New York, NY, USA: Springer, 1986, pp. 396–404.

[8] P. Vadakkepat, K. Chen Tan, and W. Ming-Liang, "Evolutionary artificial potential fields and their application in real time robot path planning," in *Proc. Congr. Evol. Comput. (CEC)*, Sep. 2000, pp. 256–263.

[9] V. Roberge, M. Tarbouchi, and G. Labonté, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 132–141, Feb. 2012.

[10] T. Fraichard, "Trajectory planning in a dynamic workspace: A 'state-time space' approach," *Adv. Robot.*, vol. 13, no. 1, pp. 75–94, Jan. 1999.

[11] F. Large, C. Laugier, and Z. Shiller, "Navigation among moving obstacles using the NLVO: Principles and applications to intelligent vehicles," *Auto. Robots*, vol. 19, no. 2, pp. 159–171, Sep. 2005.

[12] A. S. Matveev, M. C. Hoy, and A. V. Savkin, “A globally converging algorithm for reactive robot navigation among moving and deforming obstacles,” *Automatica*, vol. 54, pp. 292–304, Apr. 2015.

[13] T. Lolla, P. F. J. Lermusiaux, M. P. Ueckermann, and P. J. Haley, “Time-optimal path planning in dynamic flows using level set equations: Theory and schemes,” *Ocean Dyn.*, vol. 64, pp. 1373–1397, Oct. 2014.

[14] T. Lolla, P. J. Haley Jr., and P. F. J. Lermusiaux, “Path planning in multiscale ocean flows: Coordination and dynamic obstacles,” *Ocean Model.*, vol. 94, pp. 46–66, Oct. 2015.

[15] A. Wu and J. P. How, “Guaranteed infinite horizon avoidance of unpredictable, dynamically constrained obstacles,” *Auto. Robots*, vol. 32, no. 3, pp. 227–242, Apr. 2012.

[16] Y. M. H. Xiao and Y. Pan, “Time-optimal path planning in an evolving ocean wave field based on reachability theory,” *IEEE Trans. Control Syst. Technol.*, vol. 31, no. 4, pp. 1957–1965, Dec. 2023.

[17] O. Bokanowski, N. Forcadel, and H. Zidani, “Reachability and minimal times for state constrained nonlinear problems without any controllability assumption,” *SIAM J. Control Optim.*, vol. 48, no. 7, pp. 4292–4316, Jan. 2010.

[18] K. Margellos and J. Lygeros, “Hamilton–Jacobi formulation for reach–avoid differential games,” *IEEE Trans. Autom. Control*, vol. 56, no. 8, pp. 1849–1861, Aug. 2011.

[19] O. Bokanowski and H. Zidani, “Minimal time problems with moving targets and obstacles,” *IFAC Proc. Volumes*, vol. 44, no. 1, pp. 2589–2593, Jan. 2011.

[20] T. Lolla, “Path planning in time dependent flows using level set methods,” M.S. thesis, Dept. Mech. Eng., Massachusetts Inst. Technol., Boston, MA, USA, 2012.

[21] M. Bardi and I. Capuzzo, *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*. Norwell, MA, USA: Birkhäuser, 1997.

[22] C.-W. Shu, “Numerical experiments on the accuracy of ENO and modified ENO schemes,” *J. Sci. Comput.*, vol. 5, no. 2, pp. 127–149, Jun. 1990.

[23] C.-W. Shu and S. Osher, “Efficient implementation of essentially non-oscillatory shock-capturing schemes,” *J. Comput. Phys.*, vol. 77, no. 2, pp. 439–471, Aug. 1988.

[24] S. Osher and R. Fedkiw, *The Level Set Methods and Dynamic Implicit Surfaces*. New York, NY, USA: Springer-Verlag, 2002.

[25] G. Russo and P. Smereka, “A remark on computing distance functions shock-capturing schemes,” *J. Comput. Phys.*, vol. 163, no. 1, pp. 51–67, 2000.

[26] I. M. Mitchell, “The flexible, extensible and efficient toolbox of level set methods,” *J. Sci. Comput.*, vol. 35, no. 2, pp. 300–329, 2008.

[27] I. Mitchell, “Application of level set methods to control and reachability problems in continuous and hybrid systems,” Ph.D. thesis, Dept. Elect. Eng., Stanford Univ., Stanford, CA, USA, 2002.

[28] D. L. Chopp, “Computing minimal surfaces via level set curvature flow,” *J. Comput. Phys.*, vol. 106, no. 1, pp. 77–91, May 1993.

[29] R. Takei, R. Tsai, H. Shen, and Y. Landa, “A practical path-planning algorithm for a simple car: A Hamilton-Jacobi approach,” in *Proc. Amer. Control Conf.*, 2010, pp. 6175–6180.

[30] J. Boissonnat and S. Lazard, “A polynomial-time algorithm for computing a shortest path of bounded curvature amidst moderate obstacles,” *Int. J. Computat. Geom. Appl.*, vol. 13, pp. 189–229, Nov. 2003.

[31] P. Zaspel and M. Griebel, “Solving incompressible two-phase flows on multi-GPU clusters,” *Comput. Fluids*, vol. 80, pp. 356–364, Jul. 2013.

[32] M. Chen, S. L. Herbert, M. S. Vashishtha, S. Bansal, and C. J. Tomlin, “Decomposition of reachable sets and tubes for a class of nonlinear systems,” *IEEE Trans. Autom. Control*, vol. 63, no. 11, pp. 3675–3688, Nov. 2018.

[33] T. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Hoboken, NJ, USA: Wiley, 2011.

[34] T. Ogilvie, “Recent progress toward the understanding and prediction of ship motions,” in *Proc. 5th Symp. Nav. Hydrodyn.*, Bergen, Norway, 1964, pp. 3–79.

[35] K. Hasselmann et al., “Measurements of wind-wave growth and swell decay during the joint north sea wave project (JONSWAP),” *Deutschen Hydrographischen Zeitschrift*, vol. 8, pp. 1–95, Jan. 1973.

[36] D. G. Dommermuth and D. K. P. Yue, “A high-order spectral method for the study of nonlinear gravity waves,” *J. Fluid Mech.*, vol. 184, pp. 267–288, Nov. 1987.

[37] M. Onorato et al., “Freely decaying weak turbulence for sea surface gravity waves,” *Phys. Rev. Lett.*, vol. 89, no. 14, Sep. 2002, Art. no. 144501.

[38] W. Xiao, Y. Liu, G. Wu, and D. K. P. Yue, “Rogue wave occurrence and dynamics by direct simulations of nonlinear wave-field evolution,” *J. Fluid Mech.*, vol. 720, pp. 357–392, Apr. 2013.

[39] G. Wang and Y. Pan, “Phase-resolved ocean wave forecast with ensemble-based data assimilation,” *J. Fluid Mech.*, vol. 918, p. 357, Jul. 2021.

[40] M. G. Crandall and P.-L. Lions, “Viscosity solutions of Hamilton-Jacobi equations,” *Trans. Amer. Math. Soc.*, vol. 277, no. 1, pp. 1–42, 1983.

[41] M. G. Crandall, L. C. Evans, and P.-L. Lions, “Some properties of viscosity solutions of hamilton-jacobi equations,” *Trans. Amer. Math. Soc.*, vol. 282, no. 2, pp. 487–502, 1984.

[42] E. N. Barron and H. Ishii, “The Bellman equation for minimizing the maximum cost,” *Nonlinear Anal., Theory, Methods Appl.*, vol. 13, no. 9, pp. 1067–1090, Sep. 1989.

[43] H. H. Johnson, “An application of the maximum principle to the geometry of plane curves,” *Proc. Amer. Math. Soc.*, vol. 44, no. 2, pp. 432–435, 1974.

[44] M. Bardi and I. Capuzzo, *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations* (Modern Birkhäuser Classics). Boston, MA, USA: Birkhäuser, 2008.