

Article

A Global Path Planning Method for Unmanned Sailboats with Optimal Time Consumption

Yong Cai ^{1,2,3}, Zhaozhao Zhang ^{4,*}, Yixin Yang ^{1,2}, Zhexuan Xu ⁴, Xing Gao ⁴, Shuo Liu ^{3,4}  and Tao Wang ⁴ 

¹ School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China; caiyong888@163.com (Y.C.); yxyang@nwpu.edu.cn (Y.Y.)

² Shaanxi Key Laboratory of Underwater Information Technology, Northwestern Polytechnical University, Xi'an 710072, China

³ Ocean Research Center of Zhoushan, Zhejiang University, Zhoushan 316021, China; shuoliu@zju.edu.cn

⁴ Ocean College, Zhejiang University, Zhoushan 316000, China; zhexuan_xu@163.com (Z.X.); xinggao@zju.edu.cn (X.G.); twang001@zju.edu.cn (T.W.)

* Correspondence: zzhao2326@gmail.com

Abstract

Unmanned sailboats rely on wind and solar energy for long-duration marine missions, yet their navigation performance is strongly constrained by sail aerodynamics, hull hydrodynamics, and time-varying wind fields. Due to their underactuated nature and the critical influence of sailing speed on safety and efficiency, this paper addresses the time-optimal global path planning problem for unmanned sailboats in complex marine environments. A customized velocity prediction program (VPP) is developed to accurately estimate sailing speeds under different true wind angles, explicitly accounting for the drift angle effect. A time-efficient cost function is constructed and integrated into the Rapidly exploring Random Tree Star (RRT*) algorithm. To improve computational efficiency, three strategies, including goal-biased sampling, artificial potential field guidance, and adjacent domain expansion, are introduced. The proposed hybrid algorithm imposes no restrictions on path step size or direction, ensuring asymptotic optimality while complying with sailboat dynamic constraints. Comparative simulations and validation tests using multiple VPP datasets demonstrate that the method is feasible, efficient, and widely applicable for unmanned sailboat navigation.

Keywords: unmanned sailboat; global path planning; improved RRT* algorithm; velocity prediction program; complex marine environment



Academic Editor: Zhiguang Feng

Received: 28 May 2026

Revised: 28 June 2026

Accepted: 6 July 2026

Published: 17 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)

[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The ocean covers roughly 70% of the Earth's surface and stands as one of the most critical natural resources sustaining human society and ecological balance. Despite its unparalleled vastness and ecological significance, merely 5% of the global marine environment has been systematically explored and thoroughly studied to date [1,2]. In this context, marine robotic systems serve an irreplaceable role in advancing large-scale, long-duration ocean exploration and scientific research initiatives. Conventional marine robots [3] are predominantly propelled by electric motors or internal combustion engines, yet their operational endurance and mission range are severely constrained by the inherent limitations of on-board energy storage devices, hindering their deployment for prolonged open-ocean missions [4].

As a specialized category of marine autonomous surface vehicles (ASVs), unmanned sailboats harness wind energy as their primary propulsion source, with supplementary power supplied by solar panels and small-scale wind turbines to support long-endurance, self-sustaining operations [5–9]. Owing to these unique advantages, unmanned sailboats have been extensively deployed in long-term oceanographic research and maritime surveillance tasks, spanning persistent marine environmental data collection [10–12], marine mammal population monitoring and behavioral research [13], and coastal harbor security as well as maritime border patrol operations [14].

Path planning represents a core enabling technology that empowers autonomous surface vehicles to execute observation, patrol, and diverse operational missions safely, efficiently and autonomously [15]. Zhou et al. [16] distilled four fundamental questions to define the path planning problem for unmanned surface vehicles: What is the current operational state? What navigable trajectories are available? What is the optimal target route? How to execute the trajectory tracking effectively? Existing global path planning methodologies for marine ASVs primarily fall into three categories: heuristic search algorithms, potential field algorithms, and swarm intelligence optimization algorithms. Heuristic search algorithms include the classical A* algorithm [17], smoothed A* algorithm [7], and Theta* algorithm [18]; potential field methods encompass the artificial potential field algorithm [19] and fast marching method [20]; and representative swarm intelligence techniques include particle swarm optimization [21], genetic algorithms [22], and ant colony optimization [23].

Unmanned sailboats constitute a highly specialized class of autonomous surface vehicles whose sailing speed is jointly determined by three core factors: the ambient environmental wind field, the inherent hydrodynamic and aerodynamic design characteristics of the vessel, and the selected sailing heading. When applying the aforementioned general path planning algorithms to unmanned sailboats, it is imperative to fully incorporate the unique dynamic constraints and sailing characteristics specific to wind-powered vessels, as generic algorithms fail to account for these specialized operational attributes.

The speed polar plot [24] provides an intuitive graphical representation of the optimal sailing speed of a sailboat across varying true wind angles, laying a foundational theoretical basis for sailboat motion control and targeted path planning. The Velocity Made Good (VMG) algorithm [25] is a dedicated local path planning method tailored for unmanned sailboats, enabling efficient and safe navigation toward target waypoints under steady, uniform wind field conditions. However, traditional sailboat velocity prediction models commonly neglect the impact of drift angle, which creates a notable discrepancy between the vessel's controlled heading and its actual trajectory direction. This oversight can induce substantial path deviations, and in severe cases, may trap the sailboat in a windward no-sail zone where forward propulsion is completely infeasible, compromising mission success and navigational safety.

Saoud developed a sailboat-adapted global path planning algorithm by integrating the Probabilistic Roadmap (PRM) and Dijkstra algorithms [26], yet this approach requires individual path segments to be shorter than the grid resolution and only guarantees probabilistic completeness rather than global optimality. The improved A* algorithm proposed by Langbein [27] and the Q-learning-based method presented by Silva Junior [5] share a critical common limitation: both generate paths on grid-based maps by restricting each node's neighbor set to the eight adjacent grid cells, imposing rigid constraints on path segment length and feasible directions. Consequently, these methods tend to yield suboptimal local solutions instead of the true globally optimal path. To mitigate this limitation in complex marine environments, Shen proposed a hybrid fusion algorithm combining improved adaptive ant colony optimization and a rolling window mechanism [3], which demonstrates

enhanced performance in minimizing total path length. Nevertheless, unmanned sailboats are anisotropic autonomous platforms, and Du empirically demonstrated that path length and sailing duration are not always positively correlated [28]. Notably, when sailing up-wind, reducing the total path length below a critical threshold may trigger a drastic increase in total voyage time, rendering short-length paths inefficient in practical applications.

Existing path planning methods fail to generate time-optimal trajectories for unmanned sailboats. Moreover, traditional velocity prediction models ignore the drift angle, which significantly reduces navigation accuracy and operational safety. To tackle these critical issues, this work develops a tailored velocity prediction program (VPP) for a prototype unmanned sailboat, explicitly incorporating drift angle effects to enhance motion prediction accuracy. For complex marine environments with static obstacles and steady non-uniform wind fields, a time-optimized cost function is formulated and integrated into the RRT* algorithm to generate minimum-time global paths. The baseline RRT* is further enhanced via goal-biased sampling, artificial potential field guidance, and adjacent domain expansion. Unlike grid-based methods, the proposed hybrid algorithm imposes no restrictions on path step size or direction, achieving asymptotic optimality while satisfying the sailboat's dynamic constraints. Comparative simulations with benchmark algorithms and validations across diverse VPP datasets confirm the method's feasibility, effectiveness, and general applicability.

The rest of this paper is organized as follows. Section 2 details the development and implementation of the drift angle-aware VPP for the sailboat prototype. Section 3 presents the framework of the hybrid improved RRT* for time-optimal global path planning. Section 4 reports simulation results and comprehensive performance analyses. Finally, Section 5 concludes the study and outlines future research directions.

2. Performance Prediction of the Sailboat Prototype

2.1. Dynamic Model Construction and VPP Formulation

VPP for sailboats typically involves first confirming the dynamical model of the sailboat [29,30] and then balancing the forces or moments in each degree of freedom to achieve the optimal sailing speed for a stable heading [31]. Figure 1 illustrates the technical approach used in this paper to derive the dynamical model for the rigid-wing unmanned sailboat. The aerodynamic model of the rigid wing was obtained through aerodynamic simulation. To determine the hydrodynamic model of the sailboat, we also conducted parameter identification experiments. To simplify the complexity of velocity prediction, this paper selects a three-degree-of-freedom model that includes longitudinal, lateral, and yaw motions, based on a simplified version of the MMG model developed by Masuyama & Fukasawa [32].

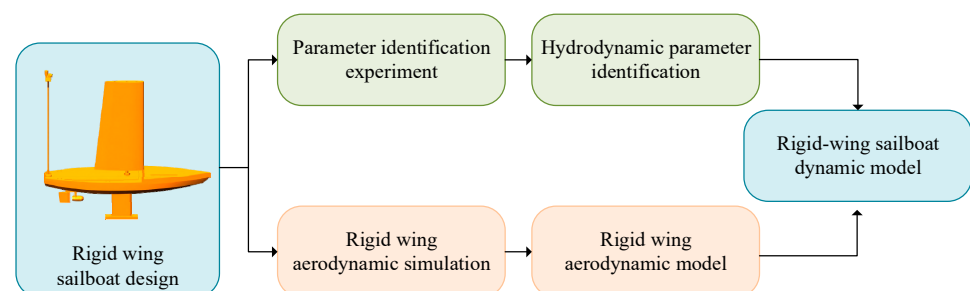


Figure 1. Technology roadmap for obtaining a dynamic model of a rigid-wing unmanned sailboat.

To formulate the velocity prediction problem, a three-degree-of-freedom dynamic model is established for the rigid wing unmanned sailboat. The model considers the surge, sway, and yaw motions in the body-fixed coordinate system. The resultant surge force,

sway force, and yaw moment are decomposed into the contributions from the hull, rudder, and rigid wing sail. Based on the physical symmetry of hydrodynamic forces acting on the hull, the longitudinal force in polynomial hydrodynamic modeling is expanded using even-order terms due to its symmetry with respect to the positive and negative drift angles. Conversely, the lateral force and yaw moment exhibit antisymmetric characteristics, as their directions of action reverse with the drift angle, and are therefore strictly expanded using odd-order terms. To balance dynamic accuracy with the computational efficiency required for global path planning, the kinematic and dynamic behaviors of the unmanned sailboat are simplified to a 3-DOF model (surge, sway, and yaw). This simplification is predicated on specific operating conditions: the environment is assumed to have moderate wind speeds (e.g., below 8 m/s, Beaufort scale 4) and relatively calm sea states (Sea State 2 or below). Under such conditions, the roll (heel) and trim angles are constrained to a small range. While large roll angles inevitably alter the effective sail area and hydrodynamic coefficients, restricting the operational envelope allows the 3-DOF model to reliably approximate the primary motion trajectory without suffering from the state-space explosion associated with a full 6-DOF model. Furthermore, within a standard hierarchical control architecture, this 3-DOF formulation serves the high-level path planner, whereas the regulation of transient dynamics, including roll and trim, is delegated to the low-level stabilizing controllers. Based on the simplified MMG formulation, the dynamic equations can be written as follows:

(1) Surge force equation

$$\begin{cases} F_x = X_u + X_h + X_r + X_s \\ X_u = -(R_f + R_r) = -\left(\frac{0.075\rho_w u^2 S_w}{2(\lg Re - 2)^2} + f(F_r)\right) \\ X_h = (X_{vv} \sin^2 \beta + X_{vvv} \sin^4 \beta) \left(\frac{1}{2}\rho_w V_b^2 LD\right) \\ X_r = C_{x\delta} \sin \alpha_r \sin \delta_r (-\rho_w V_b^2 LD) \\ X_s = C_X \left(\frac{1}{2}\rho_a V_{aw}^2 S_A\right) \end{cases} \quad (1)$$

(2) Sway force equation

$$\begin{cases} F_y = Y_h + Y_r + Y_s \\ Y_h = (-Y_v \sin \beta - Y_{vv} \sin^3 \beta) \left(\frac{1}{2}\rho_w V_b^2 LD\right) \\ Y_r = C_{y\delta} \sin \alpha_r \cos \delta_r \left(\frac{1}{2}\rho_w V_b^2 LD\right) \\ Y_s = C_Y \left(\frac{1}{2}\rho_a V_{aw}^2 S_A\right) \end{cases} \quad (2)$$

(3) Yaw moment equation

$$\begin{cases} M_z = M_h + M_r + M_s \\ M_h = (-M_v \sin \beta - M_{vv} \sin^3 \beta) \left(\frac{1}{2}\rho_w V_b^2 L^2 D\right) \\ M_r = C_{M\delta} \sin \alpha_r \cos \delta_r \left(\frac{1}{2}\rho_w V_b^2 L^2 D\right) \\ M_s = C_Y \frac{x_{CE}^G}{\sqrt{S_A}} \left(\frac{1}{2}\rho_a V_{aw}^2 S_A^{\frac{3}{2}}\right) \end{cases} \quad (3)$$

The definitions of the main variables and coefficients in the above equations are summarized in Table 1.

Table 1. Definitions of the variables and coefficients in the dynamic model.

Symbol	Definition
F_x	Resultant surge force acting on the sailboat
F_y	Resultant sway force acting on the sailboat
M_z	Resultant yaw moment acting on the sailboat
X_u	Longitudinal resistance component in the surge direction
X_h	Hull hydrodynamic force component in the surge direction
X_r	Rudder-induced force component in the surge direction
X_s	Aerodynamic force component generated by the rigid wing sail in the surge direction
Y_h	Hull hydrodynamic force component in the sway direction
Y_r	Rudder-induced force component in the sway direction
Y_s	Aerodynamic force component generated by the rigid wing sail in the sway direction
M_h	Hull hydrodynamic moment component in the yaw direction
M_r	Rudder-induced moment component in the yaw direction
M_s	Aerodynamic moment component generated by the rigid wing sail in the yaw direction
R_f	Frictional resistance of the hull
R_r	Residual resistance of the hull
ρ_w	Water density
ρ_a	Air density
u	Forward velocity of the sailboat
V_b	Boat speed
V_{aw}	Apparent wind speed
L	Waterline length of the sailboat
D	Draft of the sailboat
S_w	Wetted surface area of the hull
S_A	Reference area of the rigid wing sail
β	Drift angle of the sailboat
α_r	Rudder angle of attack
δ_r	Rudder angle
Re	Reynolds number
F_r	Froude number
$f(F_r)$	Fitted function describing the relationship between residual resistance and Froude number
X_{vv}	Hull hydrodynamic coefficient related to the sway-induced surge force
X_{vvvv}	Higher-order hull hydrodynamic coefficient related to the sway-induced surge force
Y_v	Hull hydrodynamic coefficient related to the sway force
Y_{vvv}	Higher-order hull hydrodynamic coefficient related to the sway force
M_v	Hull hydrodynamic coefficient related to the yaw moment
M_{vvv}	Higher-order hull hydrodynamic coefficient related to the yaw moment
$C_{x\delta}$	Rudder force coefficient in the surge direction
$C_{y\delta}$	Rudder force coefficient in the sway direction
$C_{M\delta}$	Rudder moment coefficient in the yaw direction
C_X	Aerodynamic force coefficient of the rigid wing sail in the surge direction
C_Y	Aerodynamic force coefficient of the rigid wing sail in the sway direction
x_{CE}^G	Longitudinal coordinate of the center of effort of the rigid wing sail in the body-fixed coordinate system with the center of gravity as the origin

According to the above formulation, the aerodynamic coefficients of the rigid wing sail and the hydrodynamic parameters of the sailboat prototype need to be determined before solving the VPP problem. In this study, the aerodynamic coefficients of the rigid wing sail are obtained through CFD simulation, while the hydrodynamic coefficients are

identified from free navigation experiments. A summary of the CFD simulation setup is presented in Table 2.

Table 2. Summary of CFD Simulation Setup.

Parameter	Configuration/Value
Mesh Type	Poly-Hexcore Hybrid Grid
Mesh Resolution	Approximately 1.5 million units
Domain Configuration	Standard wind tunnel (single inlet + single outlet)
Inlet (Inflow)	Velocity Inlet (uniform incoming wind speed)
Outlet (Outflow)	Pressure Outlet
Wing Sail Surface (Wall)	No-slip Wall
Far-field Boundaries	Top, bottom, and side boundaries (to eliminate boundary effects)

Figure 2a displays the three-dimensional model diagram of the rigid wing, while Figure 2b illustrates the computational fluid dynamic (CFD) simulation environment of the rigid wing under the given actual wind circumstances. In this simulation, by varying the angle between the rigid wing and the input true wind, the corresponding lift and drag coefficients of the sail were obtained. By fitting the resulting functions, we can calculate the forces and moments under different sail interaction conditions.

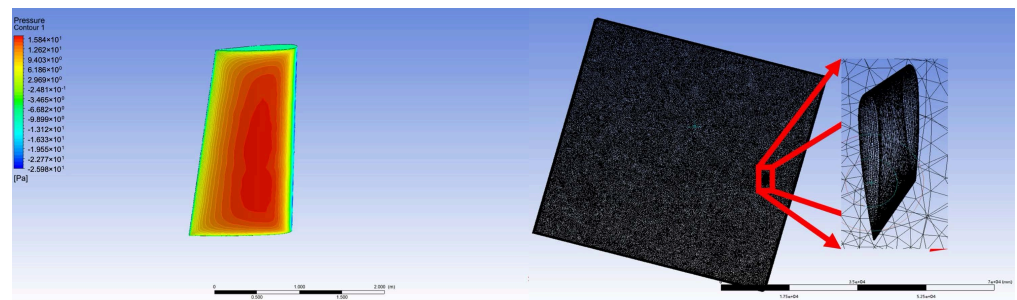


Figure 2. (a) Three-dimensional model diagram of the rigid wing. (b) Aerodynamic simulation environment for rigid wing sail.

The aerodynamic performance of the rigid wing sail was obtained from the CFD simulation. In the simulation, the lift force was defined as the aerodynamic force perpendicular to the apparent wind direction, while the drag force was defined as the aerodynamic force along the apparent wind direction. The lift and drag coefficients were calculated by nondimensionalizing the aerodynamic forces as follows:

$$C_l = \frac{2f_{ls}}{\rho_a V_{aw}^2 S_A} \quad (4)$$

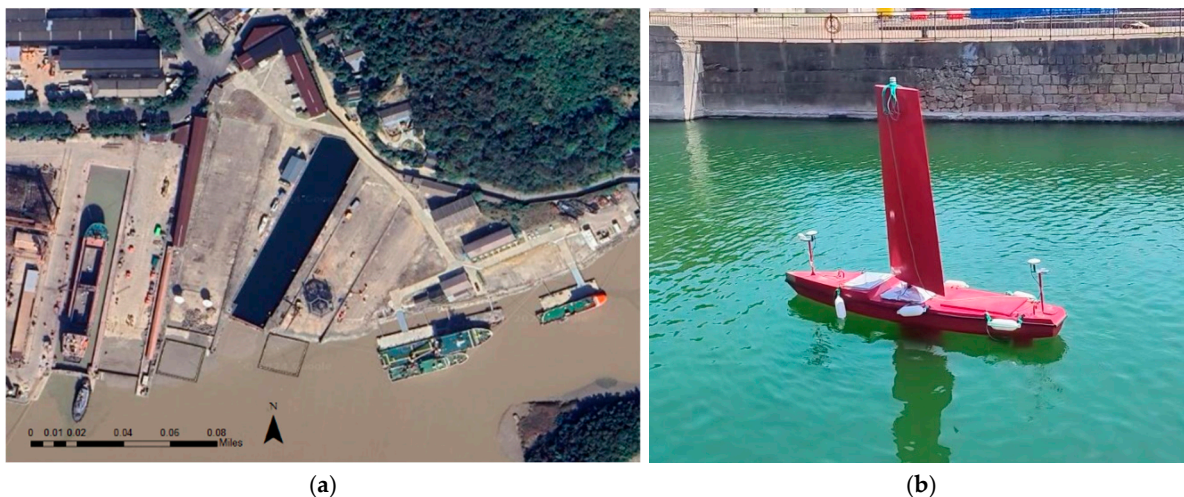
$$C_d = \frac{2f_{ds}}{\rho_a V_{aw}^2 S_A} \quad (5)$$

where f_{ls} and f_{ds} denote the lift and drag forces of the rigid wing sail, respectively; C_l and C_d are the lift and drag coefficients; ρ_a is the air density; V_{aw} is the apparent wind speed; and S_A is the reference area of the rigid wing sail. In the CFD simulation, the incoming wind speed was set to 5 m/s. The obtained aerodynamic forces and coefficients under different angles of attack are listed in Table 3.

Table 3. Lift, drag, and aerodynamic coefficients of the rigid wing sail under different angles of attack.

Angle of Attack ($^{\circ}$)	Wind Speed (m/s)	Drag Force F_{ds} (N)	Lift Force F_{ls} (N)	Drag Coefficient C_d	Lift Coefficient C_l
0	5	0.828	−0.013	0.0201	0.0000
10	5	2.514	21.797	0.0610	0.5287
20	5	9.323	25.361	0.2261	0.6152
30	5	16.525	27.946	0.4008	0.6779
40	5	24.248	29.000	0.5882	0.7034
50	5	32.181	28.174	0.7806	0.6834
60	5	39.678	24.503	0.9624	0.5944
70	5	45.185	19.009	1.0960	0.4611
80	5	48.650	11.643	1.1801	0.2824
90	5	49.647	3.491	1.2043	0.0847
100	5	48.523	−4.695	1.1770	−0.1139
110	5	45.391	−12.285	1.1010	−0.2980
120	5	40.884	−18.790	0.9917	−0.4558
130	5	36.823	−24.919	0.8932	−0.6044
140	5	33.268	−31.553	0.8070	−0.7654
150	5	15.348	−20.865	0.3723	−0.5061
160	5	9.393	−19.729	0.2278	−0.4786
170	5	3.333	−17.329	0.0808	−0.4203
180	5	1.179	−0.133	0.0286	0.0000

As illustrated in Figure 3a, we conduct free navigation sailing experiments on our sailboat in a pool situated at coordinates 122.145° E, 29.992° N. Three forms of sailing comprise free sailing: slewing, zigzag, and straight navigation. Figure 3b depicts a schematic diagram of the free sailing scenario. Based on the data obtained from experiments, we used parameter identification methods combined with the simplified MMG model to derive the dynamical model of the sailboat prototype.

**Figure 3.** (a) Site for free navigation experiment in satellite map. (b) The rigid-wing unmanned sailboat was undergoing the parameter identification experiment.

Speed prediction for the sailboat refers to predicting the maximum speed achievable when sailing towards a fixed heading under the condition of balancing forces and moments in each degree of freedom. This problem is further abstracted into a complex nonlinear constrained optimization problem in this paper and solved using a differential evolutionary algorithm [33]. Figure 4 illustrates the technical roadmap for the velocity prediction of this

sailboat presented in this paper. Regarding the speed optimization procedure illustrated in Figure 4, it is crucial to clarify the exact mathematical role of the drift angle (β). In this framework, β is neither an independent optimization variable that can be freely assigned nor an experimentally estimated constant. Rather, it is a derived quantity strictly governed by the physical equilibrium of the sailboat. For a specific wind condition and desired heading, the aerodynamic and hydrodynamic forces are highly coupled functions of both the boat speed (V) and the drift angle (β). Therefore, an inner equilibrium loop is embedded within the optimization procedure. A numerical root-finding algorithm (such as the Newton-Raphson method) iteratively adjusts both V and β until the net longitudinal force (surge) and net transverse force (sway) simultaneously converge to zero ($\sum F_x = 0$ and $\sum F_y = 0$). Once this dynamic equilibrium is satisfied, the resulting β represents the steady-state drift angle, and the corresponding V represents the optimized achievable speed.

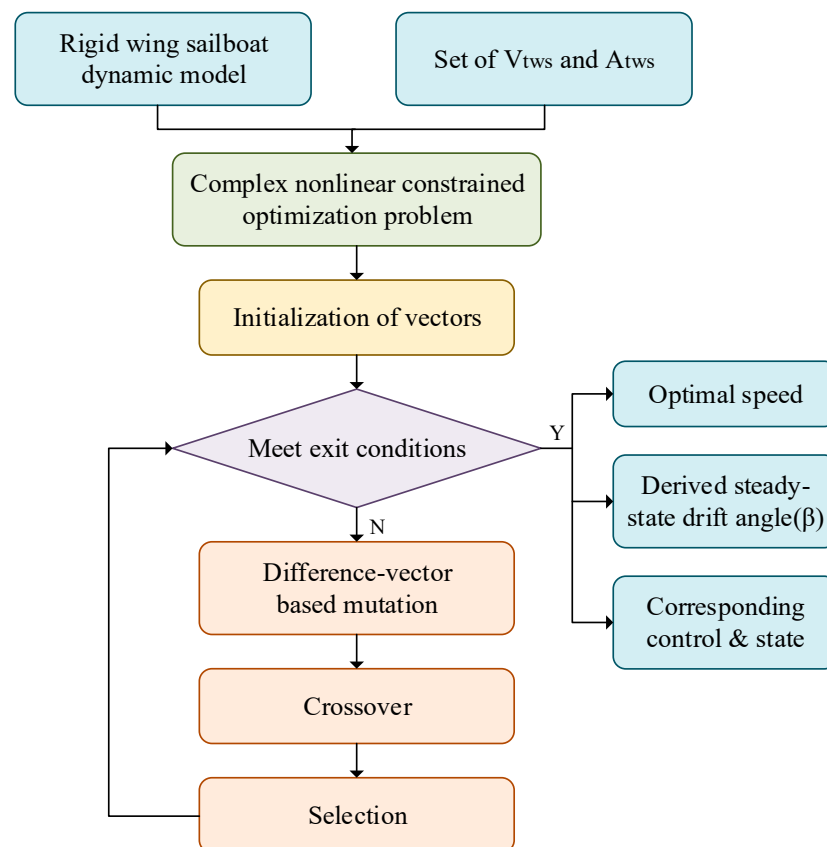


Figure 4. The technical roadmap for predicting the speed of the unmanned sailboat.

2.2. Performance Prediction Results of the Sailboat Prototype

Using the above method, we obtained the maximum steady-state speed of the sailboat at different true wind angles. The speed prediction was conducted under a constant true wind speed of 10 m/s, with the true wind angle as the input and the optimal speed as the output. The maximum achievable speed of the sailboat prototype for stable sailing at different true wind angles is shown by the red curve in Figure 5a. For true wind angles below 65° , the results either could not yield positive speeds or could not be solved, indicating that the no-sail zone of the sailboat is approximately $[-65^\circ, 65^\circ]$.

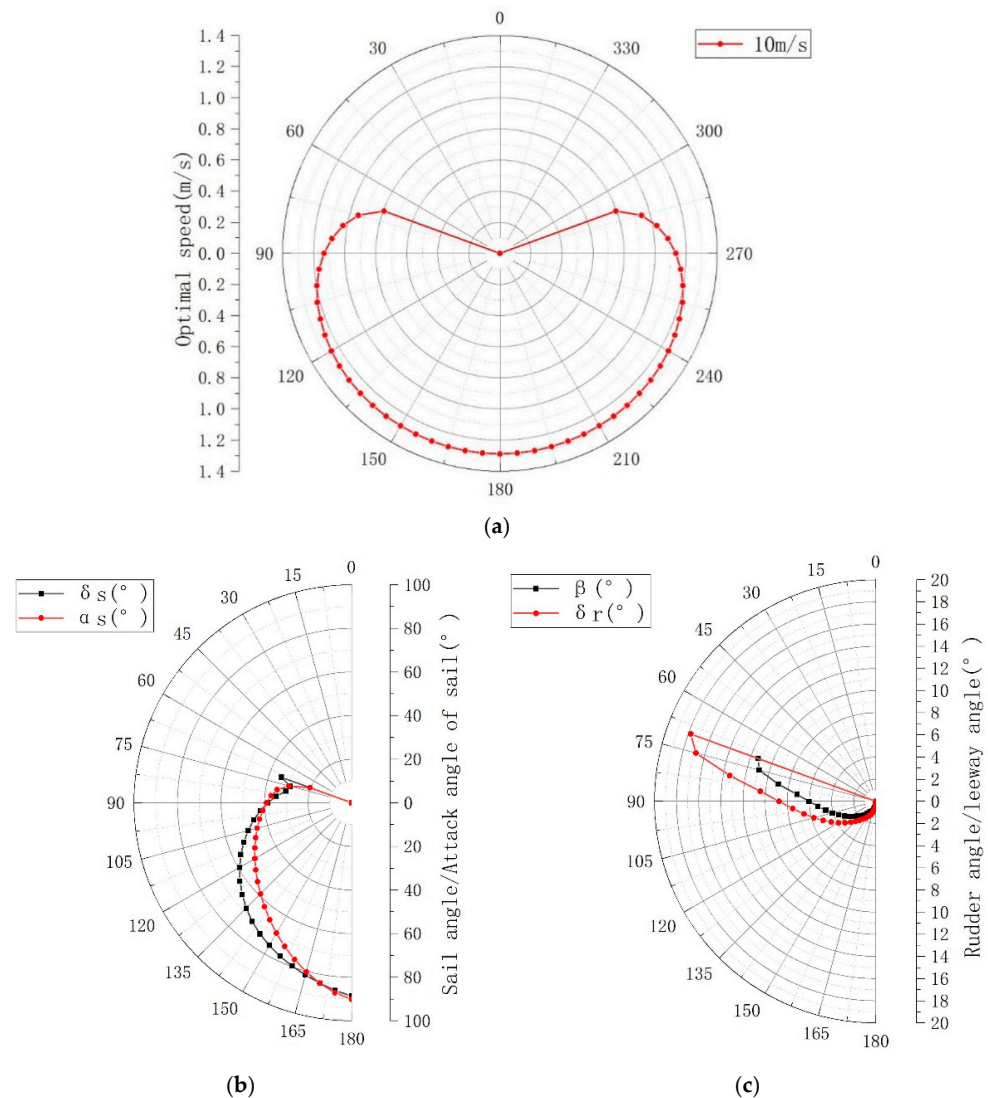


Figure 5. (a) The speed polar plot of our sailboat prototype. (b) Polar plots of the sail and attack angles corresponding to the optimal speed. (c) Polar plots of the rudder and drift angles corresponding to the optimal speed.

Figure 5b,c show the control inputs and attitudes corresponding to the maximum speed. The variables are defined as follows: δ_s represents the sail angle, α_s represents the angle of attack, β represents the drift angle, and δ_r represents the rudder angle. Figure 5b shows the variations in sail angle and attack angle with the true wind angle. As the true wind angle increases, both the sail angle and attack angle increase. When the sailboat is on a direct downwind course, the attack angle reaches 90°. Figure 5c illustrates the variations in rudder angle and drift angle with the true wind angle. It can be observed that as the sailboat approaches the no-sail zone, both the drift angle and rudder angle increase, indicating that the sailboat becomes more difficult to maintain in a stable state.

3. Improved Hybrid RRT* Algorithm

Sampling-based path planning algorithms are widely employed in robot path planning. Typical representatives include PRM [34], RRT [35] and RRT* [36–38]. The RRT* algorithm is characterized by asymptotic optimality: as the number of samples increases, the generated path asymptotically converges to the optimal path [39].

However, the motion of a sailboat is highly complex, since its propulsion is significantly affected by wind conditions, sail angles, and potentially sophisticated dynamic

characteristics. Therefore, accurate motion modeling is essential. In Section 2, we modeled and analyzed the aerodynamic and hydrodynamic performance of the sailboat prototype and obtained corresponding speed prediction results. In this section, we construct an appropriate cost function and integrate the sailboat motion model into the planning algorithm.

Nevertheless, the standard RRT* algorithm [40] suffers from low computational efficiency in steady non-uniform wind fields [41]. To address this issue, this paper introduces goal-biased sampling, potential field, and adjacent node strategies to optimize the algorithm. Furthermore, the proposed improved hybrid algorithm imposes no constraints on the step size or direction of the generated path, and it is a heuristic variant. By introducing search biases to accelerate convergence, these strategies partially sacrifice strict theoretical asymptotic optimality in exchange for enhanced computational efficiency in engineering practice.

3.1. Incorporating Sailboat Dynamical Constraints

In sailboat path planning, path length and sailing time are not always proportional, owing to the vessel's unique kinematics and the influence of the environmental wind field. In some scenarios, the shortest geometric path may lead to a substantial increase in sailing time. This indicates that taking path length as the only optimization objective is not always appropriate for sailboat path planning.

In Section 2.2, the maximum stable sailing speeds at different true wind angles were obtained using a velocity prediction program, together with the corresponding control inputs (sail angle and rudder angle) and attitude (drift angle). A suitable cost function aimed at minimizing sailing time was constructed, and the velocity prediction results were integrated into the algorithm. The improved hybrid algorithm accounts for both the kinematic constraints of the sailboat and the effects of wind (wind direction and wind speed).

Global path planning aims to find the optimal path from the start point to the target point in a known, static environmental model. In practical applications, the static obstacle information required by the planner can be acquired and updated using onboard perception sensors. For example, obstacle detection based on the fusion of 2D LiDAR and depth camera has been investigated for intelligent robots, which provides a feasible perception solution for obtaining obstacle information in real environments [42]. For unmanned sailboats, the global wind field can be acquired and updated via meteorological forecasting systems. We assume that the forecast wind field is constant within a given period and marine area and that the sailboat operates in a steady non-uniform wind field. The proposed algorithm does not impose constraints on path length or direction, which requires evaluating paths that traverse multiple wind regions. In this paper, when estimating the travel time of path segments covering multiple grids, an approximate method is adopted by accumulating the sailing times of the sub-paths within each grid.

$$CT = \sum_{i=1}^n CT^i = \sum_{i=1}^n k \frac{\text{len}(i)}{V_{boat}^i} \quad (6)$$

$$V_{boat}^i = VPP(V_{TW}^i, A_{TW}^i) = \frac{V_{TW}^i}{V_{TW}^{ref}} VPP(V_{TW}^{ref}, A_{TW}^i) \quad (7)$$

where CT is the time cost of the path, k is the resolution of the raster map, len is the length of the path, and V_{boat} is the speed of the sailboat, which is obtained through the VPP. V_{TW} is the actual speed of the wind, and the true wind angle A_{TW} is defined as the angle between the wind direction and the heading. The superscript i is the identification of the sub-path, and the superscript ref is the identification of the reference value; for

example, V_{TW}^{ref} is the reference value of the true wind speed, and in this paper, V_{TW}^{ref} is taken as ten m/s. It should be noted that the proposed cost function primarily focuses on the steady-state kinematics to evaluate global path efficiency. While transient maneuvering dynamics—such as yaw inertia and speed loss during tacking or jibing—are critical for micro-level control, they are temporarily abstracted in this high-level global planner to maintain computational efficiency.

When using speed polar plots, approximating the sailing speed through rounding can introduce errors if the true wind speed differs from the preset true wind speed. Preparing numerous sets of speed polar plots in advance can also be quite labor-intensive. Stelzer & Pröll normalized the speed polar plots and then multiplied them by the true wind speed to describe the sailing speed under any true wind speed, achieving good results. Therefore, this paper adopts a similar normalization approach [9].

Considering the motion characteristics of the sailboat, the no-sail zone is treated as an obstacle. Thus, collision detection can be represented as:

$$CT = \inf if \begin{cases} \exists A_{TW}^i \in DeadZone \\ \exists seg.len(j) \cap Obstacle \neq \emptyset \end{cases} \quad (8)$$

Algorithm 1 outlines the execution of the cost function. The algorithm requires two nodes, wind field information mapped into a grid, and speed polar plot data as inputs. The output of the algorithm is the time it takes to travel between the two places along the path.

Algorithm 1 computeCost function

```

1:   Input  $n_1, n_2, Wind, VPP$ 
2:   Output  $timeCost$ 
3:    $pathAngle \leftarrow atan(n_1, n_2)$ 
4:    $pathSegs \leftarrow pathSeg(n_1, n_2)$ 
5:   for  $i = 1$  to  $length(pathSegs)$  do
6:      $segLen_i \leftarrow pathSegs.length$ 
7:      $Atw_i \leftarrow computeAtw(pathAngle, pathSegs(i), Wind)$ 
8:      $V_{boat}^i \leftarrow VPP(V_{tw}^i, Atw_i)$ 
9:      $timeCost \leftarrow timeCost + segLen_i / V_{boat}^i$ 
10:  end for

```

3.2. Algorithm Optimization

In stable, non-uniform wind fields, considering the kinematic characteristics of the sailboat can lead to lower computational efficiency of the algorithm. To accelerate the search process, this paper introduces goal bias for sampling point selection, uses a potential field to determine new nodes, and employs an adjacent field to identify adjacent nodes for the new nodes.

During the sampling process, a certain probability is used to either directly generate the goal point or increase the sampling frequency around the goal point to expedite the expansion of the tree towards the target area. First, the algorithm sets a bias probability threshold $p_{end} \in (0, 1)$. Before sampling, the algorithm generates a random value p between 0 and 1. If $p < p_{end}$, the algorithm either sets the goal point as the sampling point or samples around the goal point. Otherwise, the algorithm performs random sampling on the global map.

In determining new nodes, the algorithm introduces attractive and repulsive potential fields to guide the expansion of the tree, optimize path quality, and improve the efficiency of path planning. The core idea is that the goal point exerts an attractive force, F_{att} , on the

new node, while obstacles exert a repulsive force, F_{rep} , on it. As shown in Figure 6, the sampling point sample generates the random direction vector step based on the step size. When generating the new node, the goal point end exerts an attractive force vector, F_{att} , while the obstacle exerts a repulsive force vector, F_{rep} . The combined vector F_{total} alters the growth direction of the new node, causing it to deviate from the direction of the sampling point towards the new node.

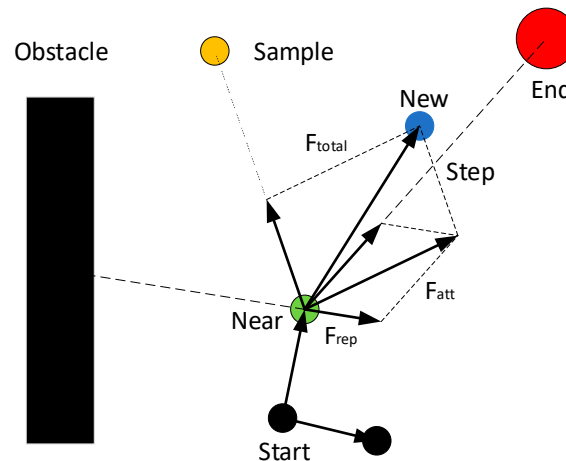


Figure 6. Guiding the construction of new nodes through the introduction of attractive and repulsive forces.

This method not only guides the path toward the endpoint but also avoids collisions with obstacles. The attractive force F_{att} and repulsive force F_{rep} can be expressed as:

$$F_{att}(x) = k_{att} \frac{(x_{end} - x_{near})}{\|x_{end} - x_{near}\|} \quad (9)$$

$$d_x = x_{near} - x_{obs} \quad (10)$$

$$F_{rep}(x) = \begin{cases} k_{rep} \left(\frac{1}{\|d_x\|} - \frac{1}{d_0} \right) \frac{1}{\|d_x\|^2} \frac{d_x}{\|d_x\|} & , \text{ if } \|d_x\| \leq d_0 \\ 0 & , \text{ if } \|d_x\| > d_0 \end{cases} \quad (11)$$

where k_{att} is the gain constant for the attractive force, x_{end} is the coordinates of the goal point, x_{near} is the node closest to the new node, $(x_{end} - x_{near})$ is the vector from x_{near} to x_{end} , and $\|x_{end} - x_{near}\|$ is the Euclidean distance between x_{near} and x_{end} . k_{rep} is the gain constant for the repulsive force, x_{obs} is the coordinates of the geometric center of the obstacle, d_x is the vector from x_{obs} to x_{near} , $\|d_x\|$ is the Euclidean distance between x_{obs} and x_{near} , and d_0 is the influence range of the obstacle.

In searching for neighboring nodes, the traditional RRT* algorithm exhaustively evaluates the cost of all nodes to the new node. In stable non-uniform wind fields, considering the sailboat kinematic constraints leads to lower computational efficiency. This paper adopts an adjacent field approach, simplifying the complex cost calculation process into a straightforward positional judgment. After normalizing the speed polar plot, the sailing speed is V_b . The radius of the neighboring domain is the product of the predicted sailing speed and the time threshold. As shown in Figure 7, the blue pie chart represents the normalized speed polar plot, and the gray area denotes the neighboring domain of node new. All tree nodes within this neighboring domain are considered neighboring nodes to new.

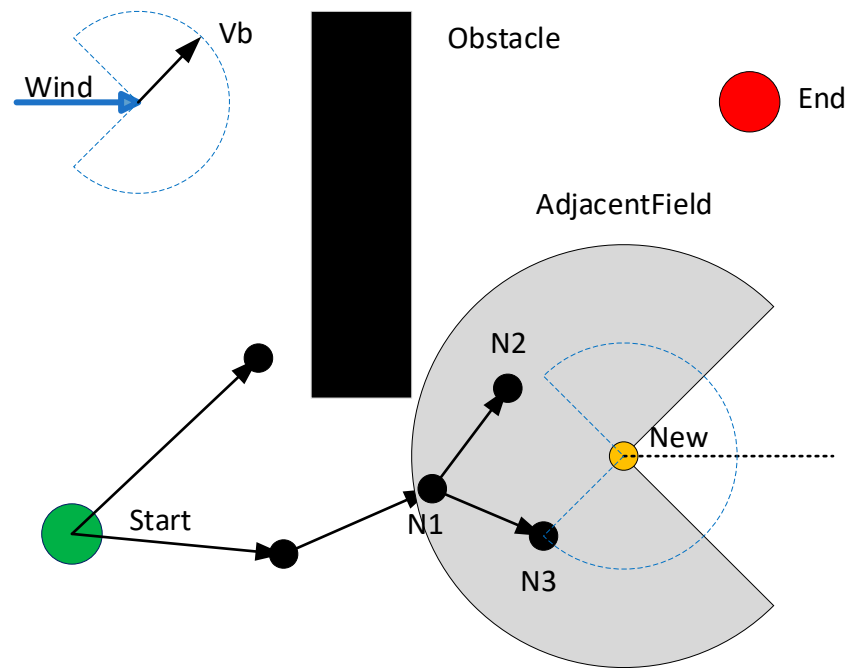


Figure 7. Finding neighboring nodes through the neighboring domain.

In summary, the unmanned sailboat path planning algorithm based on velocity prediction and hybrid RRT* is shown in Algorithm 2. The global marine environment is known, and the wind field is a stable, non-uniform field obtained from meteorological forecasting systems. We perform velocity prediction on the sailboat prototype in advance and normalize the speed polar plot. In Algorithm 2, the algorithm sets the bias probability p_{goal} to obtain the sampling point n_{sample} . The function `apfSteer` uses attractive and repulsive forces to guide the expansion of the new node n_{new} . The function `adjField` uses the neighboring domain to find neighboring nodes n_{near} and the closest node n_{min} . The function `chooseParent` reassigns the parent node for the new node. The function `insertNode` integrates the new node n_{new} into the tree T . The function `rewire` performs re-wiring of the tree T . The algorithm ultimately produces a set of path points from the start node n_{start} to the goal node n_{goal} .

Algorithm 2 VPP RRT* improved algorithm

Input: Map, Wind, VPP, n_{start} , n_{goal}

Output: A set of path points from n_{start} to n_{goal}

$T \leftarrow \text{Initialize Tree}()$

for $k = 1$ to N **do**

if $p < p_{goal}$ **then**

$n_{sample} \leftarrow n_{goal}$

else

$n_{sample} \leftarrow \text{sample}(\text{Map})$

end if

$n_{new} \leftarrow \text{apfSteer}(T, \text{Map}, n_{sample})$

$n_{min}, n_{near} \leftarrow \text{adjField}(T, n_{new}, \text{AdF}, \text{Wind}, \text{VPP})$

$n_{parent} \leftarrow \text{chooseParent}(T, n_{near}, n_{min}, n_{new}, \text{Map}, \text{Wind}, \text{VPP})$

$T \leftarrow \text{insertNode}(n_{parent}, n_{new}, T, \text{Wind}, \text{VPP})$

$T \leftarrow \text{rewire}(T, n_{near}, \text{Map}, \text{Wind}, \text{VPP})$

end for

4. Simulation Results and Analysis

To verify whether the improved hybrid RRT* algorithm can generate feasible and effective global paths for unmanned sailboats, extensive simulation experiments are carried out in this section. The experiments are divided into three parts. First, the proposed algorithm is compared and analyzed with other representative algorithms. Second, path planning is performed using different velocity prediction results to evaluate the applicability of the algorithm. Finally, the improved algorithm is implemented on electronic nautical charts and tested in a stable non-uniform wind field environment. To apply the proposed path planning algorithm to real-world marine scenarios, continuous spatial data from Electronic Navigational Charts (ENCs, e.g., standard S-57 format) must be transformed into a discrete 2D environment map. The transformation pipeline consists of four main steps. First, relevant vector features, including coastlines, depth contours, and isolated obstacles, are extracted from the ENC database. Second, a safety depth threshold is established based on the draft of the unmanned sailboat and a predefined safety margin. Areas with water depths shallower than this threshold are categorized as unnavigable regions. Third, the continuous geographic coordinates are projected into a local Cartesian system and rasterized into a discrete binary matrix, where free navigable spaces are encoded as 0 and static obstacles as 1. Finally, to simplify the collision-checking process during path optimization, the concept of Configuration Space is applied: all obstacle boundaries are inflated by a geometric radius equivalent to the sailboat's dimensions. The resulting inflated binary grid serves as the direct environmental input for the proposed VPP-based routing algorithm.

4.1. Comparison with Other Algorithms

To verify the feasibility and effectiveness of the improved algorithm for unmanned sailboats in searching for paths from the start point to the goal point, this section compares the improved hybrid RRT* algorithm with the classic RRT*, the improved APF RRT* algorithm, and the improved VPP RRT* algorithm through comparative simulation experiments. To control variables, the parameters of the algorithms were standardized. The simulation area is 100×100 . The wind field is stable and uniform, with a wind speed set at ten m/s, as shown in Figure 8. The starting point is (1, 99), marked in green, and the goal point is (99, 1), marked in red. The step size is set to 3, and the threshold is set to 10. The algorithm terminates the search as soon as a node falls within the threshold range of the goal point. It should be noted that the 100×100 rectangular simulation area utilized in this section does not represent a genuinely confined micro-sea area in the real world. Instead, it is designed as a dimensionless benchmark grid (or a scaled canonical environment). The primary objective of conducting comparative experiments in this standardized grid is to temporarily isolate the algorithm from the morphological complexities of real-world geographical data. This allows for a pure and rigorous verification of the algorithm's fundamental logic and baseline performance—specifically regarding obstacle avoidance effectiveness, path smoothness, and solving convergence—within a controlled setting. Following the validation of these core mechanisms, real-world Electronic Navigational Chart (ENC) data are subsequently introduced to evaluate the algorithm's practical applicability.

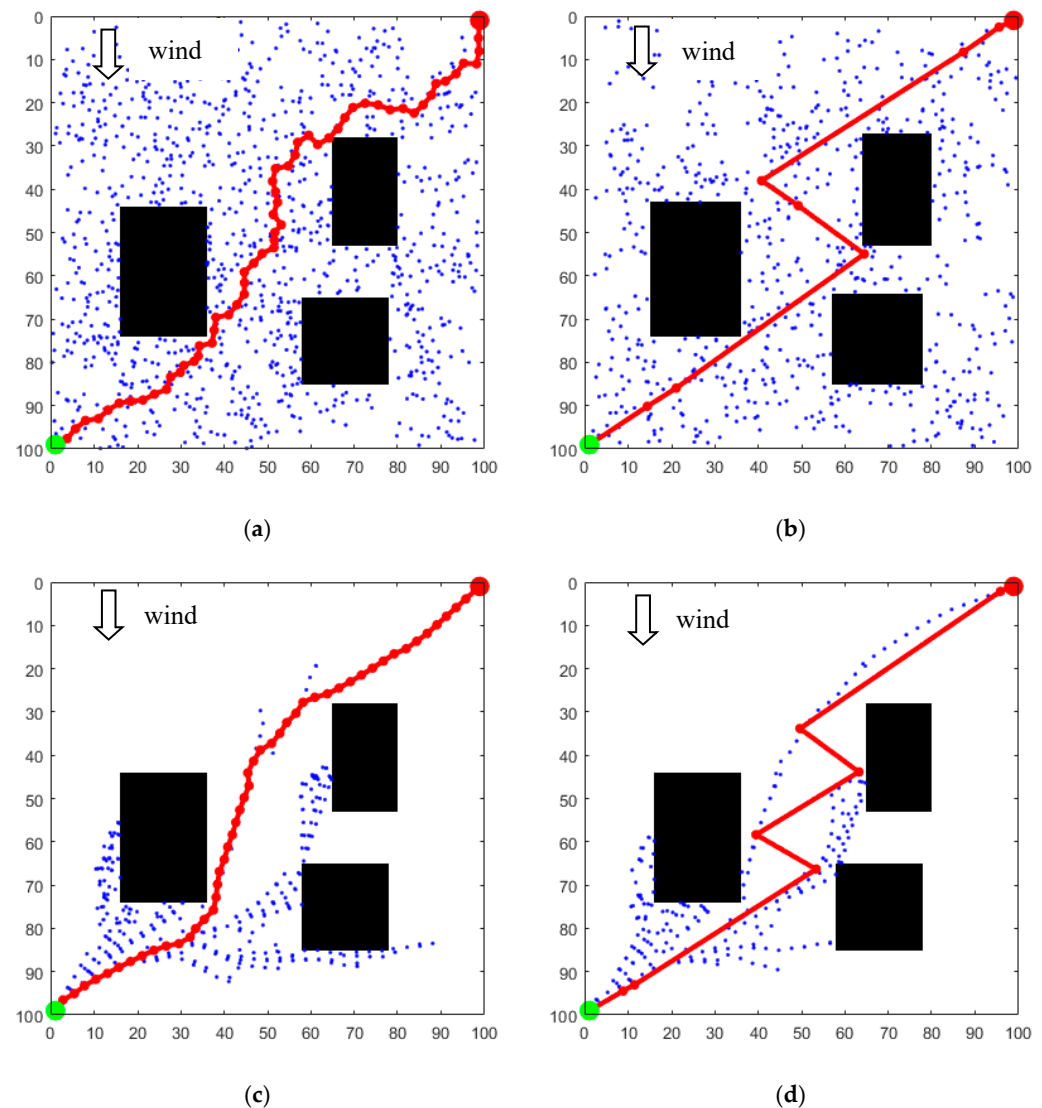


Figure 8. (a) Result of classic RRT*. (b) Result of improved APF RRT* algorithm. (c) Result of improved VPP RRT* algorithm. (d) Result of improved hybrid RRT* algorithm.

Figure 8a shows the simulation result for the classic RRT* algorithm. The blue dots represent the nodes of the search random tree, and the red line indicates the final path generated by the algorithm. Due to the kinematic constraints of the sailboat, it is unable to travel within its upwind no-sail zone. Additionally, it can be observed that in the classic RRT* algorithm, the nodes are randomly generated during the search, with no heuristic information guiding the growth of the random tree. Figure 8b shows the simulation result for the improved VPP RRT* algorithm. In this study, velocity prediction is used for the first time to enhance the RRT* algorithm. For convenience in the simulation experiments, we used velocity prediction results from an open-source project available on GitHub (Ggruszczynski, 2018, <https://github.com/ggruszczynski/VPP> (accessed on 27 May 2026)). It can be seen that the path found by this method is zigzagged, avoiding the no-sail zone of the sailboat. However, the search random tree expands randomly within the global space, and the direction of node expansion often deviates from the direction of the goal point. Figure 8c shows the simulation result for the improved APF RRT algorithm. In this study, the RRT algorithm was enhanced using the concepts of goal bias and potential field. The goal bias probability threshold was set to 0.3, the attraction gain constant k_{att} was set to 3, and the repulsion gain constant k_{rep} was set to 2. It can be seen that this method effectively

avoids ineffective searches far from the target area, allowing the random tree to expand towards the goal point more purposefully. Compared to the basic RRT* algorithm, this method significantly reduces the number of iterations and generated nodes. However, due to the constraints of the sailboat dynamics, there is still a considerable risk associated with navigating along the found path. Figure 8d shows the simulation result for the improved hybrid RRT* algorithm. It can be observed that this method finds a zigzag path that takes into account the no-sail zone of the sailboat. Additionally, by incorporating a series of optimization techniques, the search area was reduced compared to the improved VPP RRT algorithm. With global information available, the hybrid algorithm effectively balances the feasibility and effectiveness of global path planning for the unmanned sailboat.

Under the condition of uniform algorithm parameters, this section conducted 50 simulations each for the classic RRT* algorithm, the improved APF RRT* algorithm, the improved VPP RRT* algorithm, and the improved hybrid RRT* algorithm in both downwind and upwind environments. The comparison of average iterations, average search time, and average time cost is shown in Figure 9. Detailed data comparisons for both upwind and downwind conditions are provided in Tables 4 and 5. The improved hybrid RRT* algorithm trades a negligible amount of sailing time optimality for a significant enhancement in computational efficiency, which is crucial for real-time online planning. It is worth noting that in upwind conditions, due to the dynamic constraints of the sailboat, the sailboat is virtually unable to navigate the paths planned by the classic RRT* algorithm and the improved APF RRT* algorithm. In the 50 simulations conducted in this section, the sailboat could not navigate any of the paths generated by the classic RRT* algorithm and the improved APF RRT* algorithm. In downwind conditions, the sailboat is generally able to navigate the paths planned by the improved APF RRT* algorithm. This could be attributed to the guiding effects of the goal bias and potential field on the algorithm. However, due to the high randomness of the classic RRT* algorithm, there is still a 24% chance that the sailboat will enter the no-sail zone in the 50 simulations conducted in this section. The path time cost of the improved hybrid RRT* algorithm is approximately 72% of that of the basic RRT* algorithm. Considering all the experimental results, the average search time of the improved hybrid RRT* algorithm is about 40% of that of the improved VPP RRT* algorithm.

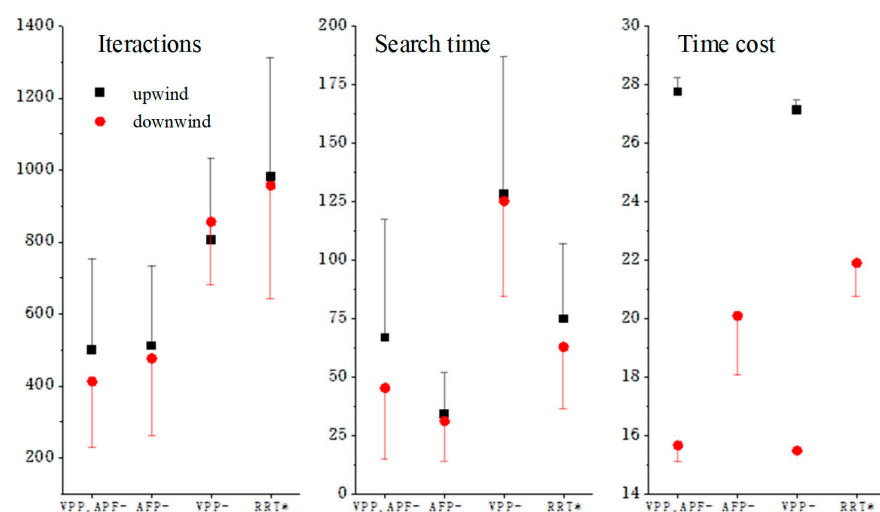


Figure 9. Data comparison from the simulation experiments (In upwind conditions, the time cost for the classic RRT algorithm and the improved APF RRT algorithm is inf; in downwind conditions, when calculating the average time cost for the classic RRT* algorithm, experimental results where the time cost is inf are not considered).

Table 4. Comparison of simulation data in upwind conditions.

Upwind	Average Time Cost	Average Iterations	Average Search Time	Feasibility
Improved VPP RRT* algorithm	27.75	500.82	66.76	1
APF RRT* algorithm	inf	511.60	34.05	0
VPP RRT* algorithm	27.15	885.36	154.71	1
Basic RRT* algorithm	inf	981.86	74.83	0

Table 5. Comparison of simulation data in downwind conditions.

Downwind	Average Time Cost	Average Iterations	Average Search Time	Feasibility
Improved VPP RRT* algorithm	15.66	413.14	45.35	1
APF RRT* algorithm	20.10	476.86	31.23	1
VPP RRT* algorithm	15.49	856.80	125.30	1
Basic RRT* algorithm	21.90	957.38	62.91	0.76

After incorporating speed prediction into the RRT* algorithm, the path cost within the sailboat no-sail zone increases significantly. Nodes tend to avoid searching within the no-sail zone, which contributes to the safe navigation of the sailboat in upwind conditions. With the introduction of goal bias, potential field, and adjacent field, the random tree is directed to grow towards the goal point, reducing the search in invalid spaces. This significantly enhances the computational efficiency of the algorithm.

4.2. Comparison with Other Speed Prediction Results

To verify the broader applicability of the improved hybrid RRT* algorithm, this section compares and analyzes different speed prediction results for various sailboats. The simulation environment is a 30×30 grid. The starting point is at (13, 29), and the destination is at (13, 1). The wind field is uniform, with a speed of 10 m/s, and the wind direction is shown in Figure 10b. The positions and sizes of obstacles are known.

Figure 10a illustrates the speed polar plots used in the improved hybrid algorithm. The blue curve represents the speed polar plot for our prototype sailboat, while the red curve represents the speed polar plot for the sailboat used in Section 4.1. The red curve has a smaller no-sail zone in the upwind direction. As shown by the red path in Figure 10b, the proposed algorithm effectively avoids the no-sail zone and static obstacles in the path it plans. In Figure 10b, the blue path shows a zigzag pattern both globally and locally. This is reasonable because the speed polar plot used has a larger no-sail zone in the upwind direction. When the upwind no-go zone of the sailboat increases, the sailboat needs to make more turns to navigate safely in upwind conditions.

It is worth noting that the no-sail zone in the speed polar plot of our prototype is $[-65^\circ, 65^\circ]$, which exceeds the range of $[-45^\circ, 45^\circ]$. In the basic A* algorithm, each sub-nodes of the grid cell are located at the centers of the cells directly above, above-right, right, below-right, below, below-left, left, and above-left, as shown in Figure 11a. In the north wind environment depicted, the sailboat no-sail zone includes sub-nodes at 225° , 270° , and 315° . These three nodes are prohibited from being reached. In the environment shown in Figure 11b, where the goal is located to the north of the starting point, an algorithm

that restricts step length and direction may not plan a feasible path for the sailboat, as the three nodes in the northern direction are prohibited from being reached. In practice, the size of the green area in Figure 11a represents the capability of a sailboat for upwind sailing. Theoretically, as long as the sailboat no-sail zone does not exceed $[-90^\circ, 90^\circ]$, the sailboat should be capable of upwind sailing. As shown by the blue path in Figure 10b, the improved hybrid RRT* algorithm can better utilize the upwind sailing capabilities of a sailboat, demonstrating a broader applicability.

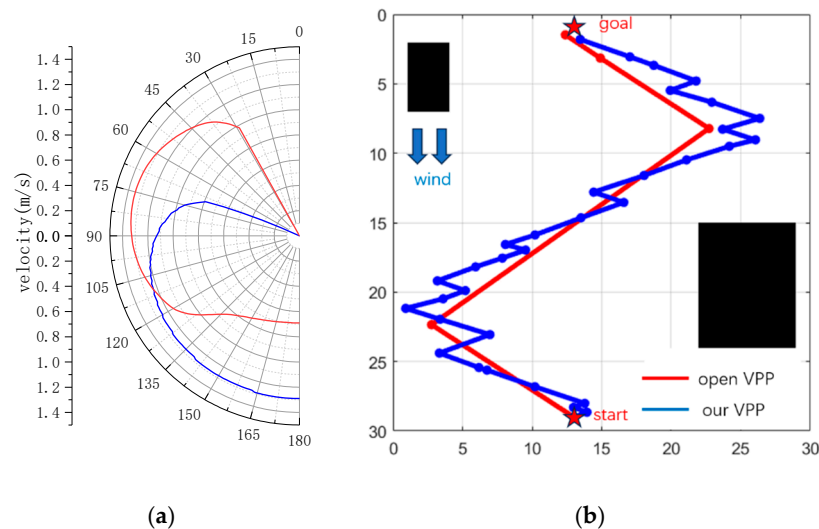


Figure 10. Data comparison from the simulation experiments (In upwind conditions, the time cost for the classic RRT algorithm and the improved APF RRT algorithm is inf; in downwind conditions, when calculating the average time cost for the classic RRT* algorithm, experimental results where the time cost is inf are not considered). Comparison of simulation results: (a) speed polar plots used in the improved hybrid algorithm; (b) planned paths obtained using the corresponding speed polar plots.

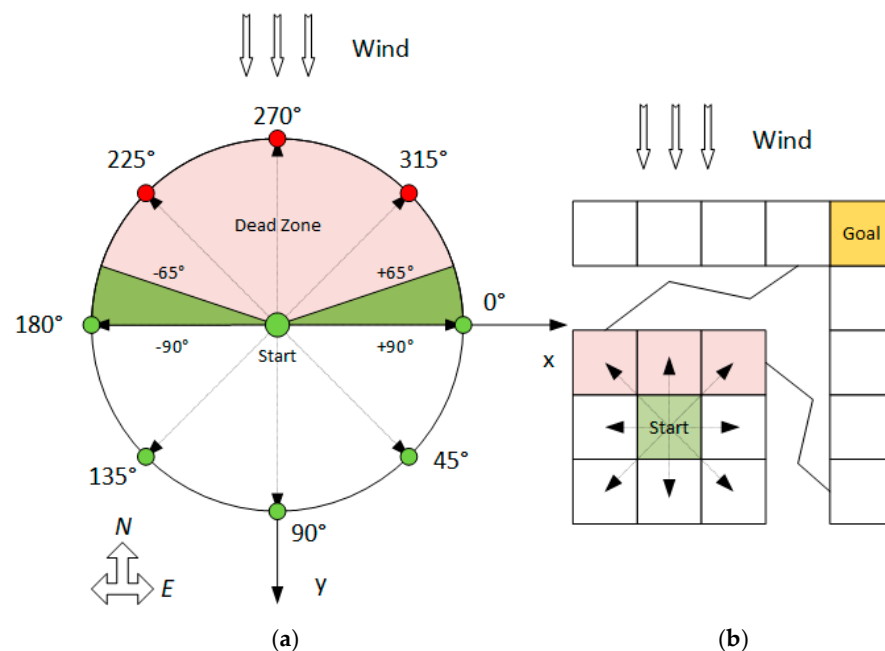


Figure 11. (a) The no-sail zone of the sailboat is $[-65^\circ, 65^\circ]$, and path points located in the no-sail zone are forbidden to be reached. (b) The sailboat cannot travel from the start to the goal in a completely northerly wind environment, as all north-facing path directions are prohibited.

5. Conclusions

This study addresses the time-optimal global path planning problem for unmanned sailboats in complex marine environments. A velocity prediction program (VPP) is developed that explicitly accounts for drift angle effects, and the corresponding speed polar diagram for the sailboat prototype is established. A time-efficient cost function is formulated to integrate the VPP into the RRT* algorithm, which is further enhanced via goal-biased sampling, artificial potential field guidance, and adjacent domain expansion.

Comparative simulations in Section 4.1 show that the proposed hybrid RRT* reduces travel time to approximately 72% of that achieved by the conventional RRT*. Additional validations are performed using different sailboat velocity prediction datasets, and the algorithm is further tested on electronic nautical charts under steady non-uniform wind fields. Results confirm that the hybrid algorithm outperforms existing methods in finding feasible, time-efficient paths with broad applicability in known marine environments.

Future work will focus on developing a local path planner that incorporates the sailboat's precise turning radius. Integrating the global and local planners will enable deployment on a rigid-wing unmanned sailboat, supporting long-endurance, long-range maritime surveillance missions.

Author Contributions: Funding acquisition, Y.C. and S.L.; Investigation, Y.C. and T.W.; Methodology, Y.C., Z.Z., Y.Y. and S.L.; Project administration, S.L.; Supervision, Y.Y.; Writing—original draft, Y.C. and Z.Z.; Writing—review and editing, Z.X., X.G. and T.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the “Pioneer” and “Leading Goose” R&D Program of Zhejiang (2023C03124 and 2022C03041).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Costanza, R. The Ecological, Economic, and Social Importance of the Oceans. *Ecol. Econ.* **1999**, *31*, 199–213. [\[CrossRef\]](#)
2. Mogias, A.; Boubonari, T.; Kevrekidis, T. Examining the Presence of Ocean Literacy Principles in Greek Primary School Textbooks. *Int. Res. Geogr. Environ. Educ.* **2021**, *30*, 314–331. [\[CrossRef\]](#)
3. Shen, Z.; Ding, W.; Liu, Y.; Yu, H. Path Planning Optimization for Unmanned Sailboat in Complex Marine Environment. *Ocean Eng.* **2023**, *269*, 113475. [\[CrossRef\]](#)
4. Tipsuwan, Y.; Sanposh, P.; Techajaronjit, N. Overview and Control Strategies of Autonomous Sailboats—A Survey. *Ocean Eng.* **2023**, *281*, 114879. [\[CrossRef\]](#)
5. Silva Junior, A.G.D.; Santos, D.H.D.; Negreiros, A.P.F.D.; Silva, J.M.V.B.D.S.; Gonçalves, L.M.G. High-Level Path Planning for an Autonomous Sailboat Robot Using Q-Learning. *Sensors* **2020**, *20*, 1550. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Silva, M.F.; Friebe, A.; Malheiro, B.; Guedes, P.; Ferreira, P.; Waller, M. Rigid Wing Sailboats: A State of the Art Survey. *Ocean Eng.* **2019**, *187*, 106150. [\[CrossRef\]](#)
7. Song, R.; Liu, Y.; Bucknall, R. Smoothed A* Algorithm for Practical Unmanned Surface Vehicle Path Planning. *Appl. Ocean Res.* **2019**, *83*, 9–20. [\[CrossRef\]](#)
8. Stelzer, R.; Jafarmadar, K. History and Recent Developments in Robotic Sailing. In *Robotic Sailing: Proceedings of the 4th International Robotic Sailing Conference*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 3–23.
9. Stelzer, R.; Pröll, T. Autonomous Sailboat Navigation for Short Course Racing. *Robot. Auton. Syst.* **2008**, *56*, 604–614. [\[CrossRef\]](#)
10. Augenstein, T.; Singh, A.; Miller, J.; Pomerenc, A.; Dean, A.; Ruina, A. Using a Controlled Sail and Tail to Steer an Autonomous Sailboat. In *Robotic Sailing 2016: Proceedings of the 9th International Robotic Sailing Conference*; Springer International Publishing: Berlin/Heidelberg, Germany, 2017; pp. 91–103.
11. Dhomé, U.; Tretow, C.; Kуттенкеулер, J.; Wängelin, F.; Fraize, J.; Fürth, M.; Razola, M. Development and Initial Results of an Autonomous Sailing Drone for Oceanic Research. In *Marine Design XIII*; CRC Press: Boca Raton, FL, USA, 2018; Volume 1, pp. 633–644.

12. Fernandes, P.C.; Marques, M.M.; Lobo, V. Barlavento-Considerations about the Design of an Autonomous Sailboat. In *Robotic Sailing 2016: Proceedings of the 9th International Robotic Sailing Conference*; Springer International Publishing: Berlin/Heidelberg, Germany, 2017; pp. 19–30.
13. Klinck, H.; Fregosi, S.; Matsumoto, H.; Turpin, A.; Mellinger, D.K.; Erofeev, A.; Stelzer, R. Mobile Autonomous Platforms for Passive-Acoustic Monitoring of High-Frequency Cetaceans. In *Proceedings of the 8th International Robotic Sailing Conference*, Cork, Ireland, 14–15 July 2016; pp. 29–37.
14. Elkaim, G.H.; Boyce, C.L., Jr. Experimental Aerodynamic Performance of a Self-Trimming Wing-Sail for Autonomous Surface Vehicles. *IFAC Proc. Vol.* **2007**, *40*, 271–276.
15. Wu, Y.; Wang, T.; Liu, S. A Review of Path Planning Methods for Marine Autonomous Surface Vehicles. *J. Mar. Sci. Eng.* **2024**, *12*, 833. [\[CrossRef\]](#)
16. Zhou, C.; Gu, S.; Wen, Y.; Du, Z.; Xiao, C.; Huang, L.; Zhu, M. The Review Unmanned Surface Vehicle Path Planning: Based on Multi-Modality Constraint. *Ocean Eng.* **2020**, *200*, 107043. [\[CrossRef\]](#)
17. Hart, P.E.; Nilsson, N.J.; Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [\[CrossRef\]](#)
18. Daniel, K.; Nash, A.; Koenig, S.; Felner, A. Theta*: Any-Angle Path Planning on Grids. *J. Artif. Intell. Res.* **2010**, *39*, 533–579. [\[CrossRef\]](#)
19. Khatib, O. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *Int. J. Robot. Res.* **1986**, *5*, 90–98. [\[CrossRef\]](#)
20. Sethian, J.A. Fast Marching Methods. *SIAM Rev.* **1999**, *41*, 199–235. [\[CrossRef\]](#)
21. Shi, Y.; Eberhart, R.C. Empirical Study of Particle Swarm Optimization. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99*; IEEE: New York, NY, USA, 1999; Volume 3, pp. 1945–1950.
22. Maulik, U.; Bandyopadhyay, S. Genetic Algorithm-Based Clustering Technique. *Pattern Recognit.* **2000**, *33*, 1455–1465. [\[CrossRef\]](#)
23. Lazarowska, A. Ship's Trajectory Planning for Collision Avoidance at Sea Based on Ant Colony Optimization. *J. Navig.* **2015**, *68*, 291–307.
24. Kerwin, J.E. *A Velocity Prediction Program for Ocean Racing Yachts*; Ocean Race Handicapping Project, The Society of Naval Architects and Marine Engineers, SNAME Report 75–17; Department of Ocean Engineering, Massachusetts Institute of Technology (MIT): Cambridge, MA, USA, 1975.
25. Klinck, H.; Stelzer, K.; Jafarmadar, K.; Mellinger, D.K. Aas Endurance: An Autonomous Acoustic Sailboat for Marine Mammal Research. In *Proceedings of the International Robotic Sailing Conference (IRSC)*, Matosinhos, Portugal, 6–12 July 2009; pp. 43–48.
26. Saoud, H.; Hua, M.D.; Plumet, F.; Amar, F.B. Routing and Course Control of an Autonomous Sailboat. In *Proceedings of the 2015 European Conference on Mobile Robots (ECMR)*, Lincoln, UK, 2–4 September 2015; pp. 1–6.
27. Langbein, J.; Stelzer, R.; Frühwirth, T. A Rule-Based Approach to Long-Term Routing for Autonomous Sailboats. In *Robotic Sailing: Proceedings of the 4th International Robotic Sailing Conference*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 195–204.
28. Du, M.; Kang, M.; Hou, C.; Xu, J. Study of Long-Term Route Planning for Autonomous Sailboat. In *Proceedings of the International Robotic Sailing Conference*, Southampton, UK, 31 August–1 September 2018.
29. An, Y.; Yu, J.; Hu, F.; Wang, Z. Towards a General Design Evaluation Tool: The Development and Validation of a VPP for Autonomous Sailing Monohulls. *Appl. Ocean Res.* **2022**, *120*, 103053. [\[CrossRef\]](#)
30. Offshore Racing Congress. *ORC VPP—Designer's Version 2020*; Offshore Racing Congress: London, UK.
31. Claughton, A. Developments in the IMS VPP Formulations. In *Proceedings of the SNAME Chesapeake Sailing Yacht Symposium*, SNAME, Annapolis, MD, USA, 29–30 January 1999; p. D021S002R001.
32. Masuyama, Y.; Fukasawa, T. Tacking Simulation of Sailing Yachts with New Model of Aerodynamic Force Variation during Tacking Maneuver. *J. Sailboat Technol.* **2011**, *2*, 1–34.
33. Das, S.; Suganthan, P.N. Differential Evolution: A Survey of the State-of-the-Art. *IEEE Trans. Evol. Comput.* **2010**, *15*, 4–31. [\[CrossRef\]](#)
34. Kavraki, L.; Latombe, J.C. Randomized Preprocessing of Configuration for Fast Path Planning. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, San Diego, CA, USA, 8–13 May 1994; pp. 2138–2145.
35. LaValle, S.M. *Rapidly-Exploring Random Trees: A New Tool for Path Planning*; Technical Report 9811; Iowa State University: Ames, IA, USA, 1998.
36. Karaman, S.; Frazzoli, E. Sampling-Based Algorithms for Optimal Motion Planning. *Int. J. Robot. Res.* **2011**, *30*, 846–894. [\[CrossRef\]](#)
37. Cui, X.; Wang, C.; Xiong, Y.; Mei, L.; Wu, S. More Quickly-RRT*: Improved Quick Rapidly-Exploring Random Tree Star Algorithm Based on Optimized Sampling Point with Better Initial Solution and Convergence Rate. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108246. [\[CrossRef\]](#)
38. Liao, B.; Wan, F.; Hua, Y.; Ma, R.; Zhu, S.; Qing, X. F-RRT*: An Improved Path Planning Algorithm with Improved Initial Solution and Convergence Rate. *Expert Syst. Appl.* **2021**, *184*, 115457. [\[CrossRef\]](#)
39. Wang, Y.; Li, J.; Zhao, S.; Su, P.; Fu, H.; Niu, H. Hybrid Path Planning for USV with Kinematic Constraints and COLREGS Based on Improved APF-RRT and DWA. *Ocean Eng.* **2025**, *318*, 120128.

40. Wang, J.; Li, J.; Song, Y.; Tuo, Y.; Liu, C. FC-RRT*: A Modified RRT* with Rapid Convergence in Complex Environments. *J. Comput. Sci.* **2024**, *77*, 102239. [[CrossRef](#)]
41. Deng, Y.; Zhao, J.; Xu, Y.; Yan, J.; Ma, R.; Im, N. Novel Path Planning of Unmanned Sailboats via Biased Sampling RRT* with Extended Rewire Mechanism. *Ocean Eng.* **2025**, *340*, 122261. [[CrossRef](#)]
42. Fan, B.; Zhao, H.; Meng, L. Obstacle Detection for Intelligent Robots Based on the Fusion of 2D Lidar and Depth Camera. *Int. J. Hydromechatronics* **2024**, *7*, 67–88. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.