applied sciences

MDPI

Article

Automatic Docking Trajectory Design-Based Time-Varying-Radius Dubins for Unmanned Surface Vessel

Meiyan Zhang [1], Wenyu Cai [2,*] and Hao Chen [2]

[1] College of Electrical Engineering, Zhejiang University of Water Resources and Electric Power, Hangzhou 310018, China
[2] College of Electronics and Information, Hangzhou Dianzi University, Hangzhou 310018, China
* Correspondence: caiwy@hdu.edu.cn

Abstract: Facing the demand of efficient automatic docking for USV, this paper proposes a new kind of iterative varying-radius Dubins curve design. Due to the special mobility of dockyards, one novel trajectory optimization problem considering time optimum is modeled first. Then, we use the ant colony optimization (ACO) algorithm to solve such a complex problem. In addition to the global exhaustive search algorithm and ACO-based solution, a novel distributed local path planning algorithm named ADTA-TRD is proposed to obtain the solution of the above optimization problem restricted by the local perception of the USV. In essence, this paper draws lessons from the idea of the circular arc segment and straight line segment in the traditional Dubins curve, and consequently, the smoothness of the derived automatic docking trajectory can be guaranteed. Finally, the performance of proposed automatic docking path planning method is verified through simulation results and an actual sea trial.

Keywords: unmanned surface vessel; automatic docking; varying-radius Dubins; trajectory design

check for updates

Citation: Zhang, M.; Cai, W.; Chen, H. Automatic Docking Trajectory Design-Based Time-Varying-Radius Dubins for Unmanned Surface Vessel. *Appl. Sci.* **2023**, *13*, 1583. https://doi.org/10.3390/app13031583

Academic Editor: Yosoon Choi

Received: 19 December 2022
Revised: 19 January 2023
Accepted: 20 January 2023
Published: 26 January 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Currently, unmanned surface vehicles (USVs) [1] are developing rapidly. They are widely used in different fields, such as marine data collection, and maritime target search and rescue [2–4]. As a new type of surface mobile platform that can reach a defined target location, USVs can complete tasks partially or fully autonomously. For example, in our project, a self-made USV is used as an unmanned rescue device at sea, which is used to rescue a drowning person and transport them back to the mobile dockyard. The well-designed diagram for the USV and its dockyard is described in Figure 1.

Path planning for USVs involves finding the shortest collision-free trajectory that reaches the destination; this has been extensively researched. Indeed, there are many scholars working in the field of path planning and tracking of USVs [5–15]. Fan et al. [5] propose a second path planning method for USVs. The method uses the nonlinear least squares method to obtain the motion constraint of USVs. Then, the method analyzes the three-point geometric relationship of the optimized path to ensure that the curvature radius of the optimized path meets the requirement. Liu et al. [6] propose a model predictive trajectory tracking method for USVs. The method uses the path following algorithm based on the Lyapunov model to follow the leader. Then, the method proposes the trajectory state estimator to ensure the followers pass through water channels. Muhovič et al. [7] propose a 3D-point cloud-based water obstacle tracking and detection method for USVs. The method uses a stereo camera to obtain a 3D-point cloud in front of the USV. Then, the method uses fast approximate semantic segmentation to filter the point cloud. Finally, the method uses a depth appearance model to track the obstacles. Wang et al. [8] propose a fuzzy trajectory control method for USVs under uncertainty. The method designs a reference route for the USV by control techniques and logical virtual vessel. Then, the method uses the online construction of fuzzy approximator to deal with the uncertainty of conditions. Shi et al. [9]

propose a USV trajectory reconstruction method based on automatic identification system. The method uses the kinematic model of USVs to eliminate the anomalous data of automatic identification system. Then, the method respectively uses empirical mode decomposition and Fermat spiral fitting for data denoising and curve fitting. Yu et al. [10] propose an integral sliding-mode adaptive neural scheme for path tracking of USVs under unknown disturbances. The scheme uses a finite-time predictor-based guidance law to ensure that the USVs converge to a predefined path. Then, the scheme designs the control law through the neural network and integral sliding-mode approach to ensure that the USVs move along a predefined path. Shin et al. [11] propose a dynamic-model-based adaptive path tracking control method for USVs. The method replaces the fully nonlinear dynamic model with a three-degree-of-freedom model. Then, the method incorporates the adaptive element of the tracking controller to improve the stability of the control process. Song et al. [12] propose a trajectory tracking fuzzy controller based on event triggering. The controller reduces the computational complexity and network communication burden by performing updates at the moment of event triggering. Path planning is a critical issue in the effective application of USVs. USV path planning is mainly divided into global and local path planning methods. Chen et al. [13] propose a USV path planning algorithm based on A* and a dynamic windows algorithm. The algorithm uses the A* algorithm to plan the global path. Then, the algorithm sets the objective function of the dynamic windows algorithm according to the environmental changes in order to avoid dynamic obstacles. Shah et al. [14] propose a USV path planning algorithm based on focused search. Based on the A* algorithm, the algorithm sets the local search area, including the best path. Then, the algorithm focuses on searching the local search region to improve the overall search efficiency. The proposed method uses a quadtree representation of the marine environment to efficiently compute the nodes of the visibility graphs. Liu et al. [15] propose a multi-USV path tracking control strategy in an uncertain environment. Based on the feedback control, the strategy designs a path tracking controller for a single USV. Then, the strategy uses a speed coordination controller to implement the formation path tracking of USVs. Although there are numerous studies on the path planning of USVs, their algorithms have different strengths and weaknesses.



Figure 1. Mechanical structure of the USV and dockyard.

Moreover, automated launch and recovery of USVs have been one of most important problems recently. Anderson et al. [16] analyze the application cases of the USV in different scenarios, such as pipeline inspection and marine animal monitoring. However, there is

little research on path planning of USV automatic docking. Esposito et al. [17] propose a dock identification algorithm based on 3D LiDAR scans. The algorithm removes data points that are not relevant to the dock. Then, the algorithm uses clustering-based image segmentation to determine the location of the dock. Yilmaz et al. [18] investigate the parallel docking problem for USVs, and propose an adaptive control algorithm. However, this method requires an expensive radar or stereo vision system, so it is difficult to implement in real-time environments. Wang et al. [19] propose a unified framework for USV path planning and tracking. The framework implements path planning by genetic algorithm. Then, the framework optimizes the path of the USV by the B-spline technique. Finally, the framework implements path tracking by deep reinforcement learning.

The traditional USV is a vehicle with constant speed and a fixed minimum turning radius, which is limited by the minimum turning radius. Therefore, the traditional USV models are often referred to as Dubins vehicles [20]. When information such as the leaving and arrival angle of the Dubins vehicle is known, the shortest path of the Dubins vehicle at two points can be calculated. Liang et al. [21] propose a Dubins-curve-based path planning algorithm for the task of USVs. The algorithm establishes the relationship model between the movement angle and velocity of the USV. Then, the algorithm uses the genetic algorithm to obtain the task execution order and path of the USV. Faigl et al. [22,23] utilize Dubins path planning to solve the Dubins traveling salesman problem (DTSP) and Dubins interval problem (DIP) for fixed-wing aircraft.

Facing the unique application scenarios, there is no path planning method for automatic docking in the existence of mobile dockyard. Our main contribution is to provide automatic docking trajectory design for unmanned surface vehicles. As far as we know, this paper is the first to investigate the path tracking problem of mobile dockyards in real sea environments.

This paper is organized as follows: Section 2 outlines the dynamic and kinematic model of the USV. Then, the section provides the mathematical model for automatic docking of USV. Varying-radius-Dubins-based trajectory design is proposed to address the above problem in Section 3. Extensive simulations and experimental results for automatic docking trajectory planning are verified in Section 4. Section 5 is the conclusion.

2. Preliminaries and System Description

The descriptions of the main notations in this paper are shown in Table 1.

Table 1. Descriptions of the main notations.

Notation	Definition
W	The width of region
L	The length of region
V_{min}	The minimal speed of USV
V_{max}	The maximal speed of USV
V_i	The speed of USV
k	Radius influence factor
R_{min}	The minimal radius of USV
R_{max}	The maximal radius of USV
R_i	Each segment's radius
$\zeta_i \in \Omega$	Each segment's in the trajectory
$\lvert\zeta_i\rvert$	Each segment's length
t_i	Each segment's time
$T(\Omega)$	Total trajectory time
n	The maximal number of iterations
m	The actual number of iterations
θ_i	The current heading of USV
h	The number of different speed selection
H	The set of minimal turning radius intervals
ϵ	Iteration termination conditions

2.1. Dynamics Model of USV

The dynamics of the USV in different coordinate systems in Figure 2 are described as follows.

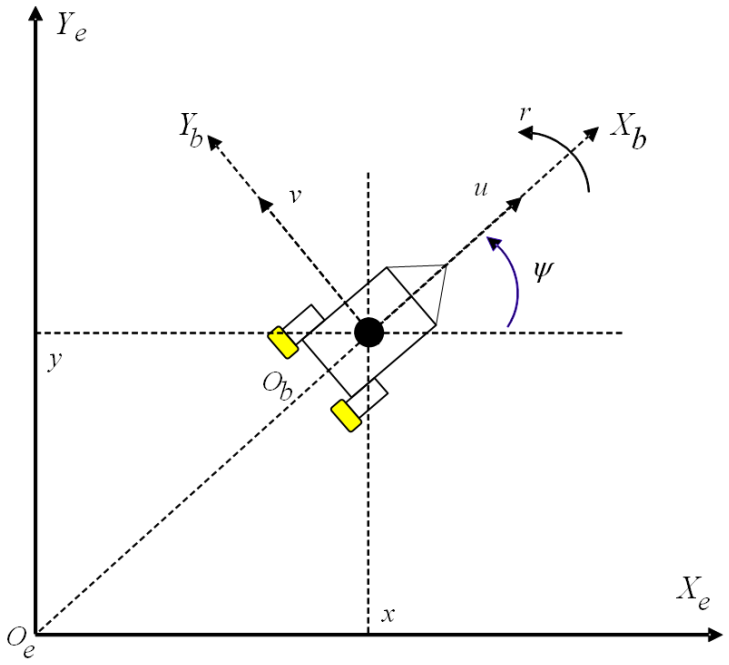

Figure 2. The dynamics and kinematics of the USV.

The standard nonlinear motion model of a well-known 3-DoF USV on the horizontal plane is described as follows. Let $\eta = [x, y, \psi]^T \in \Re^3$ denote the current position (x, y) and attitude ψ in the earth-fixed frame (inertia frame), $\omega = [u, v, r]^T \in \Re^3$ denote the surge velocity u, sway velocity v and yaw rate r in the body-fixed frame (Body frame). Then, we can establish the controlled USV dynamic system under unknown dynamics and disturbances.

$$\begin{cases} \dot{\eta} = J(\psi)\omega \\ M\dot{\omega} + C(\omega)\omega + D(\omega)\omega = \tau + \delta_d \end{cases} \tag{1}$$

where $\delta_d = [\delta_{d_u}, \delta_{d_v}, \delta_{d_r}]^T \in \Re^3$ denotes the unknown external disturbances generated by the environment, and $\tau = [\tau_u, \tau_v, \tau_r]^T \in \Re^3$ denotes the control input vector. M denotes the inertial matrix, $J(\phi)$ denotes the kinematic transformation matrix satisfying the property $J^{-1}(\psi) = J^T(\psi)$, $C(\omega) \in \Re^{3\times3}$ denotes an inertial matrix, $C(\omega) \in \Re^{3\times3}$ denotes the skew–symmetric matrix of Coriolis and centripetal terms, $D(\omega) \in \Re^{3\times3}$ denotes the damping matrix. In this paper, the η and ω are measurable and external disturbances δ_d are bound.

In this paper, the used matrices M, $D(\omega)$, $C(\omega)$ and B are defined as follows [24].

$$M = \begin{bmatrix} m_{11} & 0 & 0 \\ 0 & m_{22} & m_{23} \\ 0 & m_{32} & m_{33} \end{bmatrix} \tag{2}$$

$$D(\omega) = \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & d_{23} \\ 0 & d_{32} & d_{33} \end{bmatrix} \tag{3}$$

$$C(\omega) = \begin{bmatrix} 0 & 0 & -m_{22}v - m_{23}r \\ 0 & 0 & m_{11}u \\ m_{22}v + m_{23}r & -m_{11}u & 0 \end{bmatrix} \tag{4}$$

$$J(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

The asymmetry of the USV model causes non-zero elements in the mass inertia matrix M and damping coefficient matrix $D(\omega)$. The origin of the USV's body coordinate system is converted from the center of gravity position to the pivot point position to avoid the steering moment directly affecting the lateral speed [24]. At this stage, we transform the investigated USV model into the Formula (6) by the coordinate transformation.

$$\begin{cases} \dot{x} = u\cos\psi - v\sin\psi \\ \dot{y} = u\sin\psi + v\cos\psi \\ \dot{\psi} = r \\ \dot{u} = F_u(u,v,r) + T_u(b_{11}/m_{11}) \\ \dot{v} = X(u)r + Y(u)v \\ \dot{r} = F_r(u,v,r) + T_r(b_{32}m_{22} - b_{32}m_{23})/\left(m_{22}m_{33} - m_{23}^2\right) \end{cases} \tag{6}$$

The specific meanings of the parameters in the above equation are given in [24]. Furthermore, the state of the unmanned surface vessel η denotes the configuration (x, y, ψ). The dynamics of the USV depend on the forward speed and curvature of the trajectory κ. The curvature $\kappa = \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{\frac{3}{2}}}$ denotes a scalar value, and its sign determines whether to turn left or right in the plane. The magnitude of κ denotes constrained by the minimum turning radius R_{min} of the unmanned surface vessel. The USV's status can be expressed, as follows, by the forward speed and curvature. Therefore, under the direction, we should find the optimal path between the starting position and the ending position.

$$\dot{\eta} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = V_{usv} \begin{bmatrix} \cos\psi \\ \sin\psi \\ \kappa \end{bmatrix} \tag{7}$$

where V_{usv} and κ denote the forward speed and trajectory curvature of USV, respectively. Similar to the USV, V_{dock} and κ_{dock} denote the state related to the dockyard. Then, we define $q = [x_2, y_2, \psi_2]^T$ as the status of the dockyard. Therefore, $[q(i)]$ denotes the status of the dockyard at the i-th time stamp, which is expressed as

$$q(i+1) = q(i) + \dot{q(i)} \tag{8}$$

where $\dot{q(i)}$ is similar to the USV.

$$\dot{q} = \begin{bmatrix} \dot{x_2} \\ \dot{y_2} \\ \dot{\psi_2} \end{bmatrix} = v_{dock} \begin{bmatrix} \cos\psi_2 \\ \sin\psi_2 \\ \kappa_{dock} \end{bmatrix} \tag{9}$$

2.2. System Model

Considering the motivation given by the facts and challenges, we propose a novel algorithm for automatic docking path design. The total automatic docking system comprises of two main parts, the USV and the dockyard, as presented in Figure 3. It is worth noting that both of them are mobile, i.e., the dockyard floats with the current and the USV is guarded by its own controller. As we know, the advantage of GPS-RTK is the high positioning accuracy of a few centimeters on the surface of marine environments. Herein, the fixed and mobile GPS-RTK stations are equipped on the dockyard and USV, respectively. Moreover, two long-distance wireless transceivers are also deployed in the USV and dockyard. As a result, the relative positions and heading angles between the USV and mobile dockyard

can be measured precisely at all times. In practice, the heading value of the dockyard is synchronized to that of the USV with a long-distance wireless transceiver. The self-made RTK device is shown in Figure 4. The self-made RTK device is composed mainly of STM32 One-chip Computer, Wireless LoRa Transceiver and RTK Chip. The relative positioning accuracy can reach 10 cm, and the relative positioning distance between the USV and dockyard can reach 10 km; thus, it meets the needs of long-distance precise positioning between the USV and dockyard. On the basis of the position and heading of the mobile USV and dockyard, the optimal automatic docking trajectory can be achieved through the path planning strategy.

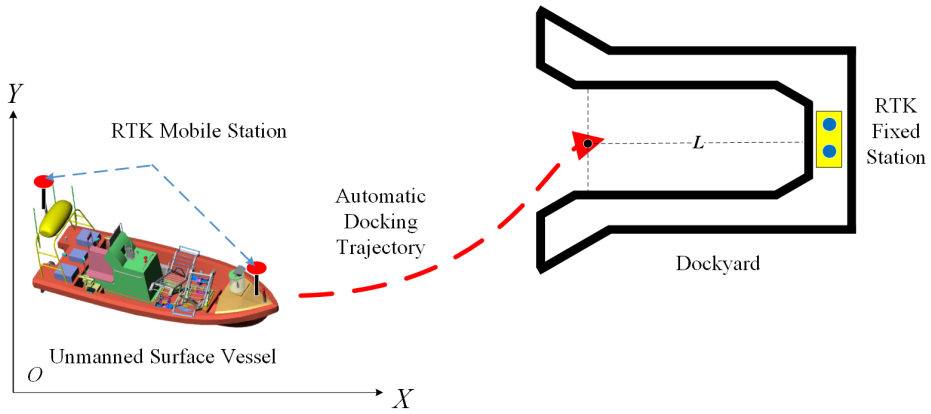

Figure 3. The framework of the automatic docking trajectory.

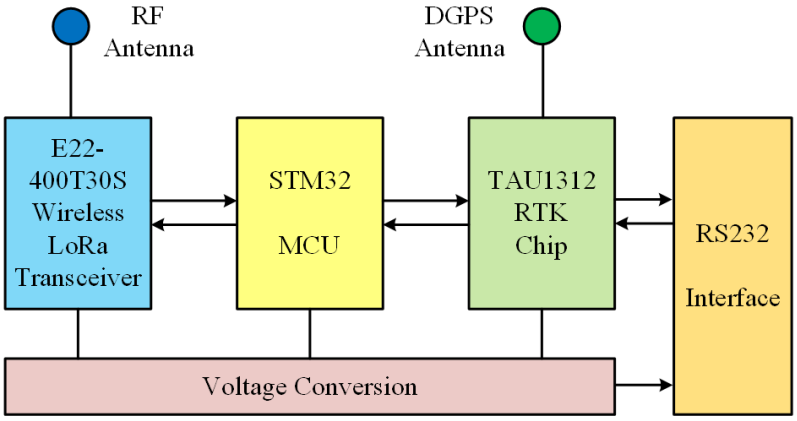

Figure 4. Self-made RTK device.

For path planning with the starting point, end point and tangent angle, to the best of our knowledge, the maximum number of possible shortest paths is six when transitioning from one Dubins state to another Dubins state [20], as shown in Figure 5 [25,26]. We can divide the six possible shortest paths into two categories based on the straight segments. The first category starts with a left (L) or right (R) turn, followed by a straight segment (S), and ending with a left or right turn (denoted by LSR, LSL, RSL and RSR, respectively). The another category uses the turning instead of straight segment (denoted by RLR and LRL). In this case, the optimal path based on the Dubins curve needs to satisfy the maximum turning angle.

In this paper, we apply the Dubins curve to design our automatic docking path. Furthermore, the traditional Dubins curve is switched to segmented mode in this paper, and each segmented Dubins curve has a different minimal turning radius. The trajectory is limited to an area of $W \times L$. Suppose that the actual speed V of the speed of the USV is limited between the minimum V_{min} and maximum V_{max} values. For simplicity, it is assumed that the speed of the USV can reach the current value, i.e., the acceleration and

deceleration times are ignored in this paper. Furthermore, it is assumed that the minimal turning radius R_i is proportional to the USV's velocity V_i under the effect of the radius influence factor k.

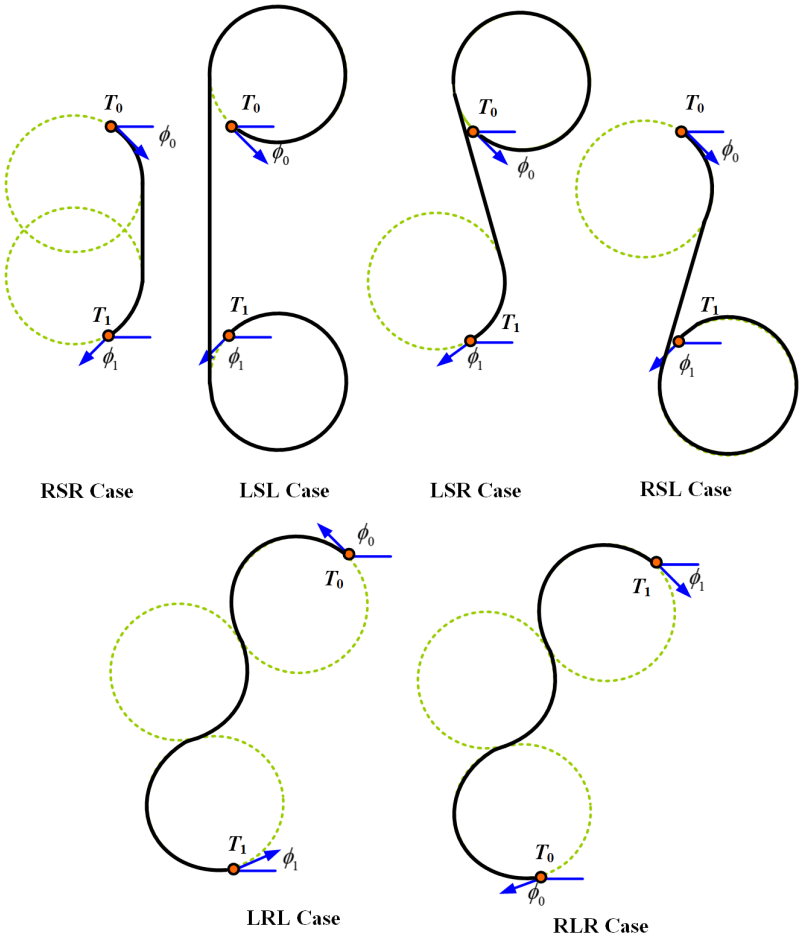

Figure 5. Six cases of the Dubins curve.

2.3. Problem Formulation

Due to external interference such as water flow, the designed dockyard will drift randomly and frequently. This paper models USV as an extended version of Dubins, that is, the speed of the USV is not constant. Therefore, the USV can change its speed to shorten the travel time. The main principle of automatic docking is to achieve target tracking in a moving scene by adjusting the USV's heading angle iteratively, which is shown in Figure 6.

The optimization objective proposed in this paper is to find a time-optimal trajectory to approach the mobile dockyard. By means of mathematical method, the optimization problem is formulated to minimize the travel time $T(\Omega)$ necessary to execute the trajectory sequence set Ω for each specific segment $\zeta_i \in \Omega, i \leq m$.

$$\begin{aligned} &\min T(\Omega) = \min \textstyle\sum_1^m t_i \\ &\textbf{s.t.:} \\ &\zeta : [0, path_{end}] \in SE(2) \\ &\zeta(0) = q(1), \zeta(path_{end}) = q(end) \\ &t_i = \tfrac{|\zeta_i|}{V_i} \\ &R_{\min} \leq kV_i \leq R_{\max} \\ &V_{\min} \leq V_i \leq V_{\max} \\ &[x, y] \in [W, L] \end{aligned} \tag{10}$$

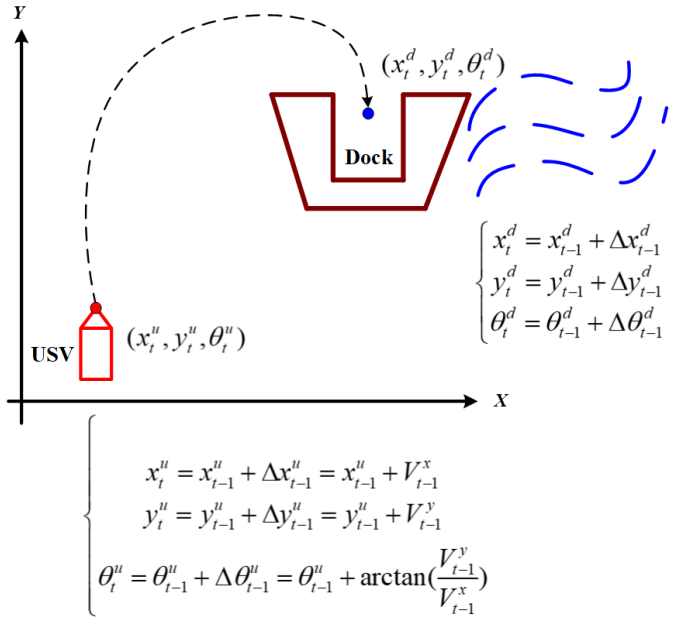

Figure 6. The principle of automatic docking design with moving dockyard.

3. Varying-Radius-Dubins-Based Trajectory Design

On the path planning level, the USV is regarded as a point-mass system, and the dynamic model of the well-designed USV is a single integrator. Therefore, each specific segment in the trajectory sequence is assumed one kind of variant Dubins curve. So far, the investigated varying-radius-Dubins-based trajectory design is to achieve the optimized path by adjusting the minimum turning radius of Dubins curve. In the other word, the proposed multi-radius Dubins path is derived from the original Dubins path with multiple segments, and each segment is designed with different minimal turning radius.

As we have seen, the key idea of our method is to sample a minimal turning radius. We consider the solution of optimal trajectory as a problem to efficiently create a set of particular minimal turning radius intervals. In Figure 7, each segment in the trajectory comprises of circle part C_i and straight line S_i, i.e., $\zeta_i = [C_i S_i]$. Since the segment length of ζ_i is $|\zeta_i|$, the total trajectory time is $T(\Omega) = \sum_1^m \frac{|\zeta_i|}{V_i}$, herein $m \leq n$. Considering the number of intervals is limited, we create an oriented graph, presented in Figure 8. Each layer corresponds to one segmented location p_i and consists of h nodes, each for one minimal turning radius in H. The graph nodes are connected by edges representing a global searching solution. Therefore, the global searching method of the time-optimal trajectory is visualized in Figure 8. Moreover, the time complexity of finding the shortest path in the graph is $O(nh^3)$.

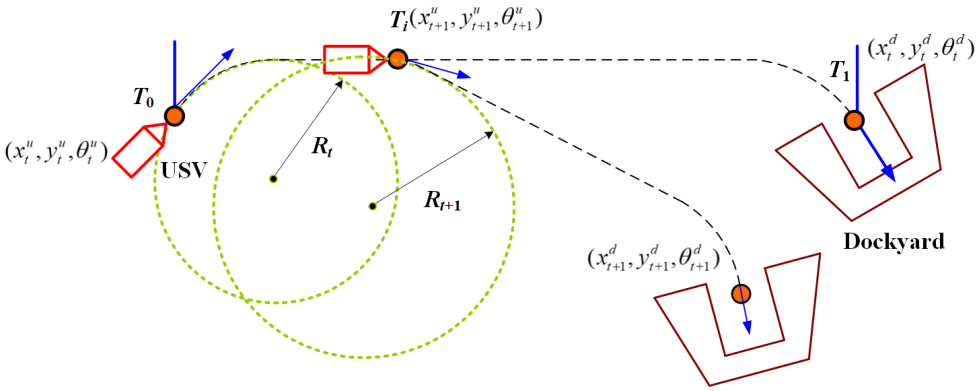

Figure 7. Varying-radius Dubins curves.

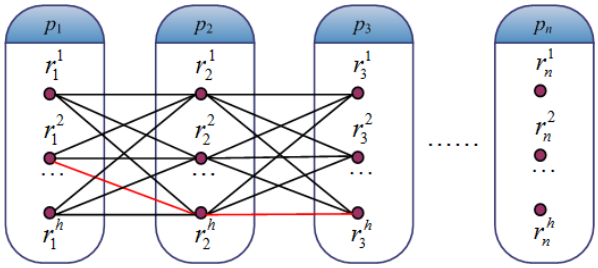

Figure 8. Global optimization searching algorithm.

Although the above global optimization method can obtain optimal solutions, it is more complex to solve. Since the heuristic method can solve the problem in a shorter time, therefore, we implement an automatic docking trajectory algorithm-based ant colony optimization (ADTA-ACO) [27] to obtain the optimal solution.

The proposed ADTA-ACO algorithm takes the sailing time of the USV as the optimization objective. As shown in Figures 9 and 10, the proposed ADTA-ACO algorithm initializes parameters such as the maximum number of iterations and the number of ant colonies. The core idea of ADTA-ACO algorithm is to select appropriate turning radius at each site step by step, and this optimization problem is solved by Ant Colony Optimization algorithm. In Figure 9, Dubins curves with different colors correspond to different turning radii. Then, ADTA-ACO algorithm takes the sailing time of the USV at different turn radii as the heuristic value. ADTA-ACO algorithm calculates the selection probability of next turn radius in combination with the pheromone concentration and heuristic value. Afterwards, ADTA-ACO algorithm selects the next turn radius until the USV returns to the dockyard according to the roulette wheel and selection probability. If all ants are traversed, the algorithm calculates sailing time of the USV for each ant. Finally, ADTA-ACO algorithm updates the global best solution and the pheromone concentration. If the maximum number of iterations is reached, ADTA-ACO algorithm outputs the global best solution as the set of minimal turning radius intervals. In conclusion, Figure 10 illustrates the main flowchart for ADTA-ACO algorithm.

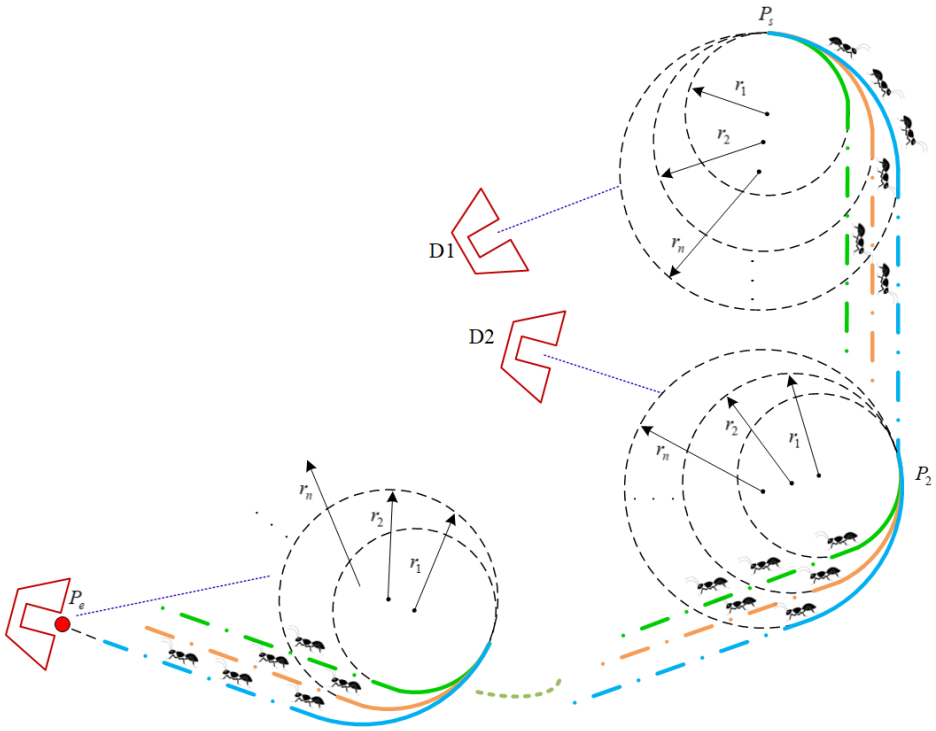

Figure 9. Automatic docking trajectory algorithm-based ant colony optimization algorithm.

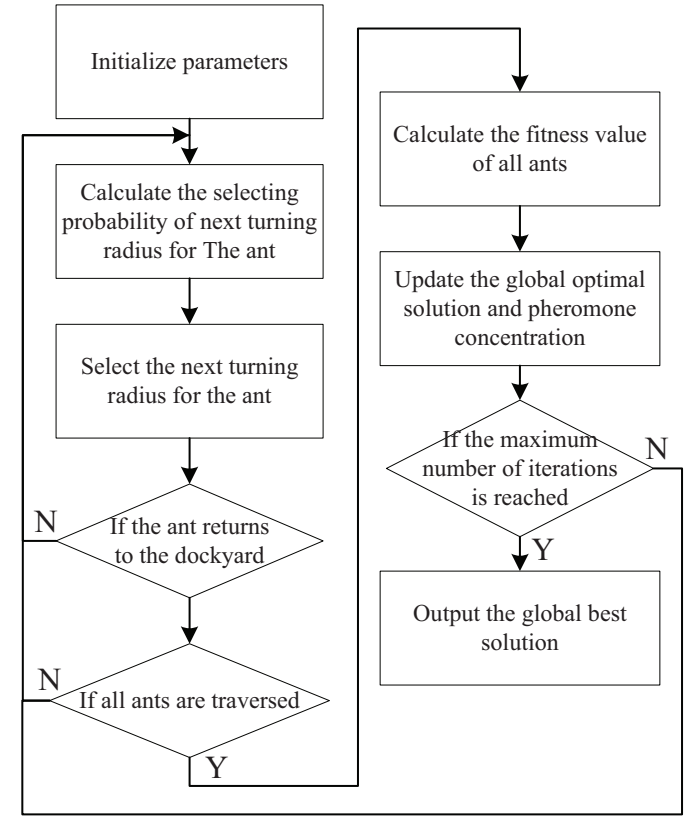

Figure 10. Main flowchart of ADTA-ACO.

Although the proposed ADTA-ACO algorithm obtains solutions close to the optimal solution in a shorter time, the USV can only perceive local information, i.e., the USV can only determine specific parameters of next segment according to the current dockyard's position and direction. Therefore, we propose an automatic docking trajectory algorithm-based time-varying-radius Dubins (ADTA-TRD), as shown in Figure 11, to calculate the optimal minimal turning radius in the next step.

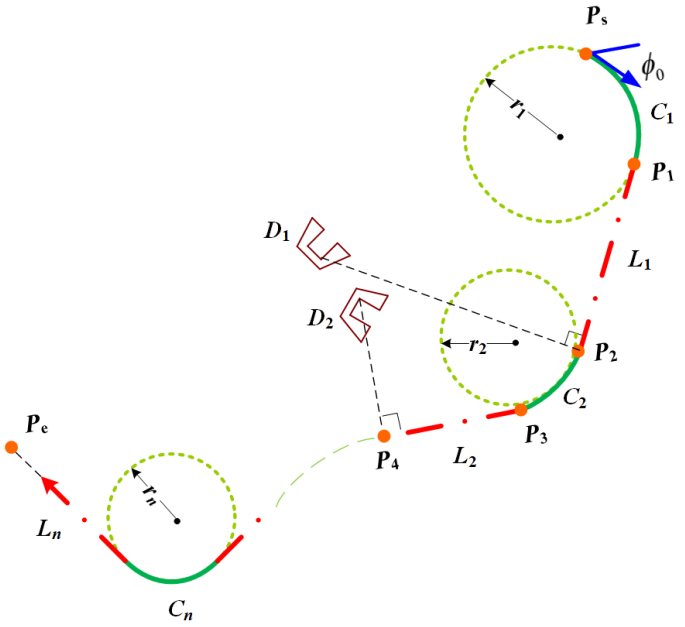

Figure 11. Distributed local path planning approach.

The proposed ADTA-TRD algorithm utilizes the Dubins path with one turn segment and one straight segment. The USV is closest to the dockyard when the straight segment of Dubins path is at an angle of 90 degrees to the current dockyard's position. Therefore, the ADTA-TRD algorithm calculates the angle between the straight segment of Dubins path and the current dock position under different turning radii. Then, the algorithm selects the turning radius closest to 90 degrees as the current optimal turning radius r_{best}. When the distance between the dockyard and the USV is less than the distance threshold D, the distributed algorithm calculates the time for the USV to reach the dockyard under different turning radii. Then, the algorithm selects the turning radius with the shortest time as the current optimal turning radius r_{best}. The core idea of the method is to optimize the minimum turning radius to obtain the fastest possible trajectory. In summary, the details of the proposed ADTA-TRD are described as Algorithm 1.

Algorithm 1 Automatic docking trajectory algorithm-based time-varying-radius Dubins (ADTA-TRD).

Input: the USV's position $(x_t^u, y_t^u, \theta_t^u)$, dockyard's position $(x_t^d, y_t^d, \theta_t^d)$, minimal speed of USV $V_{\min}$, number of different speed selection M;
Output: the set of minimal turning radius intervals H;

Initialize radius influence factor k, moving position of Dubins path $Slen = \emptyset$, end position of linear segment of Dubins path, ϵ, $ENDPoint_i = \emptyset$,...;
if Distance($x_t^u, y_t^u, x_t^d, y_t^d$)<$\epsilon$ **then** //Calculate the Euclidean distance between the USV and the dockyard.
    $f_t = 1$;
else
    $f_t = 0$;
end if
while do$f_t == 1$
    for $i = 1 \rightarrow h$ **do**
        R = f(V_i,k);//Calculate the turning radius.
        ($ENDPoint_i$, $Slen_i$, t_i) = Dubins($x_t^u, y_t^u, \theta_t^u, x_t^d, y_t^d, \theta_t^d$,$R$,$V_i$);//Calculate the Dubins path.
    end for
    if Distance(x_t^u, y_t^u,x_t^d, y_t^d)>D **then**
        for $i = 1 \rightarrow M$ **do**
            $Angle_i$=Calangle($ENDPoint_i$,x_t^u, y_t^u,x_t^d, y_t^d,V_i);//Calculate the angle between the straight segment of Dubins path and the dockyard.
        end for
        (r_{best}, V_{best})=Finda($Angle$);//Find the turning radius and speed of USV corresponding to the closest 90 degree angle.
    else
        (r_{best}, V_{best})=Findt(t);//Find the turning radius and speed of USV corresponding to the shortest time.
    end if
    $H \leftarrow r_{best}$
    ($x_{t+1}^d, y_{t+1}^d, \theta_{t+1}^d$)=Updatadockyard($x_t^d, y_t^d$,$Slen$,$V_{best}$,$r_{best}$);//Update dockyard's status.
    ($x_{t+1}^u, y_{t+1}^u, \theta_{t+1}^u$)=UpdataUSV($x_t^u, y_t^u$, $Slen$,V_{best},r_{best});//Update USV's status.
    if Distance($x_{t+1}^u, y_{t+1}^u, x_{t+1}^d, y_{t+1}^d$)<$\epsilon$ **then**
        $f_t = 1$;
    else
        $f_t = 0$;
    end if
end while

4. Simulations and Experiments

4.1. Simulation Setup

First of all, the automatic docking trajectory is verified with Matlab simulation tools. There are two kinds of dockyard movement trajectories, one of them is the translational motion following the mother ship, and the other is Eulerian-model-based motion with the movement of water. The proposed ADTA-ACO and ADTA-TRD are compared in terms of docking voyage time. The main parameters used in this simulation are listed as Table 2.

Table 2. Simulation values.

Notation	Value
k	2
V_{min}	1
V_{max}	5
R_{min}	1
R_{max}	9
W	100
L	200
n	50
θ	$\pi/4$
h	3
ϵ	5

4.2. Simulation Results

The automatic docking trajectory results of the above two proposed algorithms for translational dockyard are compared in Figure 12. The docking trajectory lengths of ADTA-ACO and ADTA-TRD, respectively, are 50.62 m and 56.82 m. The voyage times of ADTA-TRD and ADTA-ACO, respectively, are 1900.28 s and 2276.65 s. Therefore, the docking trajectory length and voyage time of ADTA-TRD is better than that of ADTA-ACO.

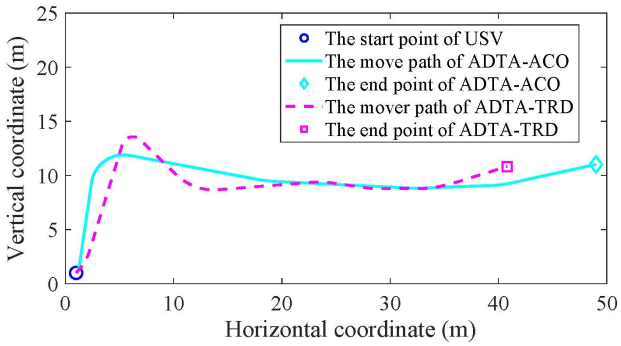

Figure 12. Automatic docking trajectories with translational dockyard.

The automatic docking trajectory results of the above two algorithms for the water-influenced mobile dockyard are compared in Figure 13. The docking trajectory length of ADTA-ACO and ADTA-TRD, respectively, are 28.4 m and 30.23 m. The voyage time of ADTA-TRD and ADTA-ACO, respectively, are 1443.65 s and 1611.6 s. Therefore, the docking trajectory length and voyage time of ADTA-TRD is better than that of ADTA-ACO.

The paper selects the radius influence factor k 1, 2, 3, 4, 5 and other simulation parameters in Table 2 for analyzing the influence of radius influence factor on voyage time. Through several round simulations, the average value of the results is obtained for comparison. As shown in Table 3, with the increase in the radius influence factor k, the docking trajectory length gradually increases, leading to a gradual increase in the voyage time of ADTA-ACO and ADTA-TRD. Meanwhile, the voyage time of ADTA-TRD is lower than that of ADTA-ACO.

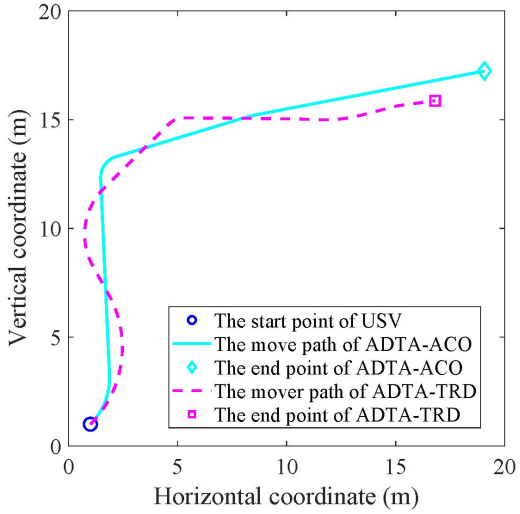

Figure 13. Automatic docking trajectories with water-influenced mobile dockyard.

Table 3. The influence of radius influence factor on voyage time.

Algorithm	k				
	1	**2**	**3**	**4**	**5**
ADTA-ACO	892.58	2292.65	3280.90	4698.95	5851.98
ADTA-TRD	722.15	1900.28	2958.53	3960.53	4890.53

The paper selects the minimal speed of the USV V_{min} 1, 2, 3, 4 and other simulation parameters in Table 2 for analyzing the influence of minimal speed of the USV on voyage time. Through several round simulations, the average value of the results is obtained for comparison. As shown in Table 4, with the increase in minimal speed of the USV V_{min}, the speed search range gradually decreases, leading to a gradual increase in the voyage time of ADTA-ACO and ADTA-TRD. Meanwhile, the voyage time of ADTA-TRD algorithm is lower than that of ADTA-ACO algorithm.

Table 4. The influence of minimal speed of the USV on voyage time.

Algorithm	V_{min}			
	1	**2**	**3**	**4**
ADTA-ACO	2176.65	2235.283	2361.63	2395.10
ADTA-TRD	1900.28	2053.65	2193.95	2249.20

4.3. Practical Sea Trial

The final practical sea trial is located in a bay of Yangjiang City, Guangdong Province, as shown in Figure 14, which is a satellite map showing the sea trial site. We take many experiments to verify the effectiveness of proposed algorithm. In this successful sea trial, a typical step in the total automatic docking trajectory is described in Figure 15. Figure 15 are shot by high-definition camera equipped on the Unmanned Aerial Vehicle, where the USV is tracking the mobile dockyard. Moreover, the final scene of the USV returning to the dockyard successfully is illustrated in Figure 16, which indicates that the automatic docking process has been completed. For clarity, the actual automatic docking trajectory on actual map is described in Figure 17. The red curve denotes the USV's trajectory, and the blue curve denotes the dockyard's trajectory. The trajectory result clearly shows that the USV and dockyard finally reached the same point. It is obvious that the USV has planned an optimal trajectory according to the Dubins curves, and successfully reached the dockyard. To summarize, the above experimental results verify the feasibility of this method.



Figure 14. Sea trial site.



Figure 15. Automatic docking process.



Figure 16. The final result of automatic docking.

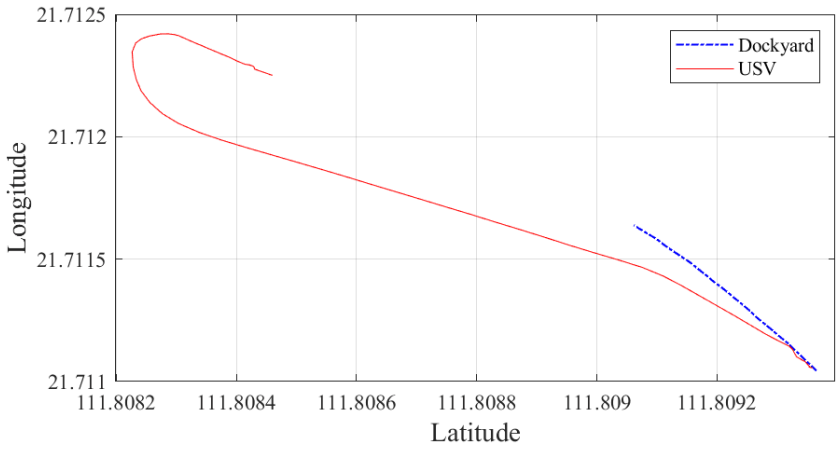

Figure 17. The actual automatic docking trajectory.

5. Conclusions

This paper proposes a form of varying-radius-based Dubins path for the USV's automatic docking design. The time optimum problems are addressed first. Moreover, a novel optimal searching algorithm is provided to solve the above optimization problem. The proposed multi-radius Dubins path provides a way to exploit the variable speed of the USV, and finds a distributed solution to the time-optimal trajectory. Finally, the performance of proposed ADTA-TRD method is verified through simulation results and an actual sea trial. To the best of our knowledge, there is no discussion in the literature about how to design tracking methods for USVs in relation to mobile dockyards. In the future, we will extend our application scenario to the cooperative operation of multiple USVs, and discuss how to achieve multi-agent collaborative optimization.

Author Contributions: Conceptualization, W.C. and M.Z.; methodology, W.C. and M.Z.; software, H.C. and M.Z.; validation, H.C. and M.Z.; formal analysis, W.C. and M.Z.; investigation, W.C. and M.Z.; resources, W.C. and M.Z.; data curation, W.C. and M.Z.; writing—original draft preparation, W.C. and H.C.; writing—review and editing, H.C. and M.Z.; visualization, M.Z.; supervision, W.C.

and M.Z.; project administration, M.Z.; funding acquisition, M.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been partially supported by Natural Science Foundation of Zhejiang Province (No.LZ22F010004 and LZJWY22E090001), National Natural Science Foundation of China (No.62271179, 61871163 and 61801431) and the Stable Supporting Fund of Acoustics Science and Technology Laboratory (JCKY2021604SSJS009).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the anonymous referees for their constructive comments and valuable suggestions, which help us improve the quality and presentation of the paper greatly.

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

References

1. Jorge, V.A.M.; Granada, R.; Maidana, R.G.; Jurak, D.A.; Heck, G.; Negreiros, A.P.F.; dos Santos, D.H.; Gonçalves, L.M.G.; Amory, A.M. A Survey on Unmanned Surface Vehicles for Disaster Robotics: Main Challenges and Directions. *Sensors* **2019**, *19*, 702. [CrossRef] [PubMed]
2. Han, J.; Kim, J. Three-Dimensional Reconstruction of a Marine Floating Structure With an Unmanned Surface Vessel. *IEEE J. Ocean. Eng.* **2019**, *44*, 984–996. [CrossRef]
3. Cheng, L.; Deng, B.; Yang, Y.; Lyu, J.; Zhao, J.; Zhou, K.; Yang, C.; Wang, L.; Yang, S.; He, Y. Water Target Recognition Method and Application for Unmanned Surface Vessels. *IEEE Access* **2021**, *10*, 421–434. [CrossRef]
4. Liu, B.; Chen, Z.; Zhang, H.-T.; Wang, X.; Geng, T.; Su, H.; Zhao, J. Collective Dynamics and Control for Multiple Unmanned Surface Vessels. *IEEE Trans. Control. Syst. Technol.* **2019**, *28*, 2540–2547. [CrossRef]
5. Fan, J.; Li, Y.; Liao, Y.; Jiang, W.; Wang, L.; Jia, Q.; Wu, H. Second Path Planning for Unmanned Surface Vehicle Considering the Constraint of Motion Performance. *J. Mar. Sci. Eng.* **2019**, *7*, 104. [CrossRef]
6. Liu, B.; Zhang, H.-T.; Meng, H.; Fu, D.; Su, H. Scanning-Chain Formation Control for Multiple Unmanned Surface Vessels to Pass Through Water Channels. *IEEE Trans. Cybern.* **2022**, *52*, 1850–1861. [CrossRef]
7. Muhovič, J.; Mandeljc, R.; Bovcon, B.; Kristan, M.; Perš, J. Obstacle Tracking for Unmanned Surface Vessels Using 3-D Point Cloud. *IEEE J. Ocean. Eng.* **2019**, *45*, 786–798. [CrossRef]
8. Wang, S.; Fu, M.; Wang, Y.; Tuo, Y.; Ren, H. Adaptive Online Constructive Fuzzy Tracking Control for Unmanned Surface Vessel With Unknown Time-Varying Uncertainties. *IEEE Access* **2018**, *6*, 70444–70455. [CrossRef]
9. Shi, B.; Su, Y.; Zhang, D.; Wang, C.; AbouOmar, M.S. Research on Trajectory Reconstruction Method Using Automatic Identification System Data for Unmanned Surface Vessel. *IEEE Access* **2019**, *7*, 170374–170384. [CrossRef]
10. Yu, Y.; Guo, C.; Yu, H. Finite-Time PLOS-Based Integral Sliding-Mode Adaptive Neural Path Following for Unmanned Surface Vessels With Unknown Dynamics and Disturbances. *IEEE Trans. Autom. Sci. Eng.* **2019**, *16*, 1500–1511. [CrossRef]
11. Shin, J.; Kwak, D.J.; Lee, Y.-I. Adaptive Path-Following Control for an Unmanned Surface Vessel Using an Identified Dynamic Model. *IEEE/ASME Trans. Mechatron.* **2017**, *22*, 1143–1153. [CrossRef]
12. Song, S.; Park, J.H.; Zhang, B.; Song, X. Event-Triggered Adaptive Practical Fixed-Time Trajectory Tracking Control for Unmanned Surface Vehicle. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *68*, 436–440. [CrossRef]
13. Chen, Z.; Zhang, Y.; Zhang, Y.; Nie, Y.; Tang, J.; Zhu, S. A Hybrid Path Planning Algorithm for Unmanned Surface Vehicles in Complex Environment With Dynamic Obstacles. *IEEE Access* **2019**, *7*, 126439–126449. [CrossRef]
14. Shah, B.C.; Gupta, S.K. Long-Distance Path Planning for Unmanned Surface Vehicles in Complex Marine Environment. *IEEE J. Ocean. Eng.* **2019**, *45*, 813–830. [CrossRef]
15. Li, Z.; Liu, Z.; Zhang, J. Multi-under-Actuated Unmanned Surface Vessel Coordinated Path Tracking. *Sensors* **2020**, *20*, 864. [CrossRef]
16. Anderson, B.S. Cost Reduction in E&P, IMR, and Survey Operations Using Unmanned Surface Vehicles. In Proceedings of the Offshore Technology Conference, Houston, TX, USA, 30 April–3 May 2018; pp. 1–10. [CrossRef]
17. Esposito, J.M.; Graves, M. An algorithm to identify docking locations for autonomous surface vessels from 3-D LiDAR scans. In Proceedings of the 2014 IEEE International Conference on Technologies for Practical Robot Applications (TePRA), Woburn, MA, USA, 14–15 April 2014; pp. 1–6.
18. Yilmaz, I.C.; Ahiska, K.; Leblebicioglu, M.K. Parallel Docking Problem for Unmanned Surface Vehicle. In Proceedings of the 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, 18-21 November 2018; pp. 744–749.
19. Wang, N.; Zhang, Y.; Ahn, C.K.; Xu, Q. Autonomous Pilot of Unmanned Surface Vehicles: Bridging Path Planning and Tracking. *IEEE Trans. Veh. Technol.* **2021**, *71*, 2358–2374. [CrossRef]

20. Dubins, L.E. On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents. *Am. J. Math.* **1957**, *79*, 497. [CrossRef]
21. Liang, X.; Jiang, P.; Zhu, H. Path Planning for Unmanned Surface Vehicle with Dubins Curve based on GA. In Proceedings of the 2020 Chinese Automation Congress (CAC), Shanghai, China, 6–8 November 2020; pp. 5149–5154. [CrossRef]
22. Váňa, P.; Faigl, J. Optimal solution of the Generalized Dubins Interval Problem: Finding the shortest curvature-constrained path through a set of regions. *Auton. Robot.* **2020**, *44*, 1359–1376. [CrossRef]
23. Faigl, J.; Vana, P.; Saska, M.; Baca, T.; Spurny, V. On solution of the Dubins touring problem. In Proceedings of the 2017 European Conference on Mobile Robots (ECMR), Paris, France, 6–8 September 2017; pp. 1–6.
24. Cheng, Y.; Zhang, W. Concise deep reinforcement learning obstacle avoidance for underactuated unmanned marine vessels. *Neurocomputing* **2017**, *272*, 63–73. [CrossRef]
25. Cai, W.; Zhang, M.; Zheng, Y.R. Task Assignment and Path Planning for Multiple Autonomous Underwater Vehicles Using 3D Dubins Curves. *Sensors* **2017**, *17*, 1607. [CrossRef]
26. Cai, W.; Zhang, M. Smooth 3D Dubins Curves Based Mobile Data Gathering in Sparse Underwater Sensor Networks. *Sensors* **2018**, *18*, 2105. [CrossRef] [PubMed]
27. Lyridis, D.V. An improved ant colony optimization algorithm for unmanned surface vehicle local path planning with multi-modality constraints. *Ocean. Eng.* **2021**, *241*, 109890. [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.