

Research Article

Reactive Path Planning Approach for Docking Robots in Unknown Environment

Peng Cui, Weisheng Yan, and Yintao Wang

School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China

Correspondence should be addressed to Weisheng Yan; wsysan@nwpu.edu.cn

Received 13 May 2017; Accepted 6 July 2017; Published 12 September 2017

Academic Editor: Cheng S. Chin

Copyright © 2017 Peng Cui et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Autonomous robots need to be recharged and exchange information with the host through docking in the long-distance tasks. Therefore, feasible path is required in the docking process to guide the robot and adjust its pose. However, when there are unknown obstacles in the work area, it becomes difficult to determine the feasible path for docking. This paper presents a reactive path planning approach named Dubins-APF (DAPF) to solve the path planning problem for docking in unknown environment with obstacles. In this proposed approach the Dubins curves are combined with the designed obstacle avoidance potential field to plan the feasible path. Firstly, an initial path is planned and followed according to the configurations of the robot and the docking station. Then when the followed path is evaluated to be infeasible, the intermediate configuration is calculated as well as the replanned path based on the obstacle avoidance potential field. The robot will be navigated to the docking station with proper pose eventually via the DAPF approach. The proposed DAPF approach is efficient and does not require the prior knowledge about the environment. Simulation results are given to validate the effectiveness and feasibility of the proposed approach.

1. Introduction

Nowadays autonomous mobile robots such as autonomous drones, autonomous underwater vehicles, and automated vehicles are widely used in the complex environment and undertake the dangerous and heavy tasks [1–8]. However, their durations are constrained by their limited battery capacities and data storage spaces [9]. To solve this problem docking stations are designed and deployed in the work areas of the robots to maintain them in practical applications [10, 11]. To ensure the safety and the success in the docking process, a feasible path is required to be planned firstly [12]. This is because, for one thing, the area for docking may be unknown in advance or dynamically changing and there may be static and moving obstacles that threaten the safety of the robots [12, 13]. For another, the final configuration (pose and velocity) of the robot should be adjusted properly to avoid the impact with the docking station. The kinematic characters of the robot should be considered as well for path tracking and energy saving [14–16].

In recent years, a variety of approaches have been proposed to solve the path planning problem in unknown

environment. Among them a biologically inspired neural network approach is presented in [17] to plan path for the autonomous underwater vehicle in unknown two-dimension (2D) environment, which is achieved via updating the environment maps according to Dempster-Shafer theory in steps. In [18] the online path planning problem with prescribed target in environment with unknown obstacles is considered and the neural networks trained by the reinforcement learning approach are adopted to solve this problem. In [19] the rapidly exploring random trees star (RRT*) algorithm is employed to plan the path for autonomous underwater vehicles, where the mutual information between the scalar field model and observations is used to improve the path planning result. In [20] the path planning problem for household robot in unknown environment is considered and solved by the modified artificial potential field (APF) method based on the motion characteristics of household animals. In [21] the collision-free path planning for autonomous container truck is achieved via utilizing the improved ant colony optimization (ACO) algorithm. In this algorithm the local path is generated according to the selected local target which is determined and updated by the rolling window

approach. In [22] a dynamic planning algorithm is presented to determine the collision-free path for the mobile terrestrial robot in unknown environment. In this algorithm the local objectives are determined by the genetic algorithm (GA) and the optimum routes are generated dynamically towards the global object. It can be concluded from the efforts above that the efficiency of the path planning approach is considered emphatically in the unknown environment. Meanwhile the reactive frameworks are adaptive in solving the path planning problem in the unknown environment in spite of the fact that different various local path planning approaches such as modified bioinspired method, RRT algorithm, and APF method are adopted. However, the smoothness of the planned path is rarely considered in these efforts as well as the pose and velocity of the robot, which is critical in the docking tasks [23].

In this paper, a reactive path planning approach named Dubins-APF is proposed to solve the path planning problem for docking in unknown environment based on combination of the Dubins curves and the APF approach. Dubins curves have been proved to be the optimal paths with minimal turning radius that connect two points with prescribed poses in 2D space [24, 25]. However, if there are obstacles in the environment for docking, it is difficult to determine the feasible path based on Dubins curves [26, 27]. Since the APF approach is efficient in obstacle avoidance, in this paper it is combined with the Dubins curves to determine the feasible path for docking [28–31]. The proposed path planning approach works in a reactive mode which is described as follows. Once the planned path is infeasible, feasible intermediate configurations are determined based on the obstacle avoidance potential field according to the configurations of the detected obstacles. Then feasible Dubins curves are generated as the replanned path based on the intermediate configuration. Through implementing this path planning-replanning strategy continually, the DAPF algorithm will solve the path planning problem for docking.

The main contributions of this paper are as follows:

- (1) The geometrical approach to determine the 3D Dubins curves is proposed in this paper. It can be utilized to generate the docking path and evaluate the feasibility of the planned path as well.
- (2) The conception of combining the advantages of the Dubins curves and the artificial potential field is proposed and implemented in this paper to improve the quality of the docking path while avoiding obstacles.

The structure of this paper is presented as follows. The path planning problem for docking and the notion of the DAPF approach are introduced in Section 1. The problem statement and the path generation approach with 3D Dubins curves are described in Section 2. In Section 3 the Dubins-APF path planning approach is proposed and illustrated in detail. The simulation result and discussion about the DAPF approach are presented in Section 4. Some conclusions and future works are provided in Section 5.

2. Path Planning with 3D Dubins Curves

2.1. Problem Statement. The path planning problem for docking is to determine a collision-free path to connect the initial position of the autonomous robot and the docking station with prescribed poses under certain constraints of the robot [28]. In this paper the docking station with unidirectional entrance is considered and it is assumed to be static. The position of the docking station is written as P_d and the direction of its entrance is expressed as \vec{V}_d . Meanwhile the velocity of the autonomous robot is assumed to be constant and its position and velocity are written as P_r and \vec{V}_r , where $\|\vec{V}_r\| = v$. The turning ability of the robot is assumed to be limited and the minimal turning radius is written as R_t . Hence to avoid collision with the docking station, the final position and velocity of the robot should be close to P_d and \vec{V}_d . Additionally the work area for docking is assumed to be unknown in advance, which means the autonomous robot can only acquire the environment information within its sensor range R_s .

2.2. 3D Dubins Curves. The Dubins curve only consists of two kinds of segments which are the circle (C) segment and the straight-line (S) segment, where the radius of the C segment is equal to the minimal turning radius of the robot and this curve is smooth at the intersections of the adjacent segments. In 2D environment the optimality of the two Dubins curves has been proved and the CCC curve or the CSC curve is reckoned as the shortest path [24]. However, the determination of the Dubins curve becomes complex in 3D environment due to the increase of the dimensionality, which means not all the segments of the Dubins curves are coplanar [32, 33]. Therefore considering the efficiency of its application in path planning, a 3D Dubins curve determination approach is presented in this paper based on the geometric characters of the typical CSC curve.

The typical CSC curve is shown in Figure 1. The initial configuration of position and pose (green vector) and the final configuration (red vector) are presented as $[P_1, \vec{V}_1]$ and $[P_2, \vec{V}_2]$ and the feasible path that connects them is a 3D Dubins curve which consists of one S segment and two C segments. The C segments C_1 (red arc) and C_2 (green arc) are two circular arcs with centers O_1 and O_2 and radii R_1 and R_2 , respectively. They are connected by the S segment (blue straight line) with intersections P_5 and P_6 separately. To indicate the coplanarity of the Dubins curve, two auxiliary lines L_1 and L_2 are drawn as Figure 1 shows. L_1 has P_1 on it and parallels with \vec{V}_1 and L_2 has P_2 on it and parallels with \vec{V}_2 . They intersect with the elongation of the S segment at P_3 and P_4 , respectively. Based on the spatial relations of L_1 , L_2 , and the S segment, the segments of the CSC curve can be divided into two planes determined by L_1 and the S segment and L_2 and the S segment separately. The intersection line of these two planes is the S segment.

Inspired by the geometric relations of these two planes, the determination process of the 3D Dubins curve is implemented as follows. The intersections of L_1 , L_2 , and the elongation of S are presented as

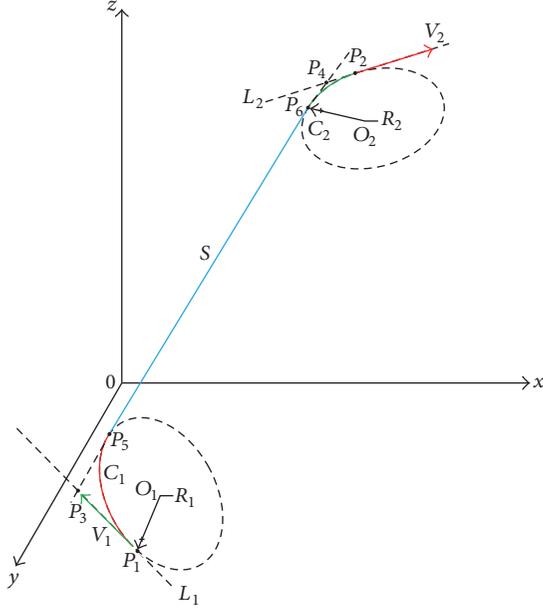


FIGURE 1: The illustration of the typical 3D CSC curve.

$$\begin{aligned} P_3 &= P_1 + k_3 \vec{V}_1, \\ P_4 &= P_2 + k_4 \vec{V}_2, \end{aligned} \quad (1)$$

where k_3 and k_4 are nonzero constants. A special case of (1) should be noticed where either k_3 or k_4 is infinite. It means that the corresponding C segment is semicircular arc and this case will be discussed later. Once k_3 and k_4 are determined, the intersections of C_1 and C_2 with S can be expressed as

$$\begin{aligned} P_5 &= P_3 + k_5 \vec{V}_3, \\ P_6 &= P_4 + k_6 \vec{V}_3, \end{aligned} \quad (2)$$

where k_5 and k_6 are constants and \vec{V}_3 is the vector that indicates the moving direction of the robot on the S segment. Since the Dubins curve is smooth at the intersections of every two adjacent segments, \vec{V}_3 can be obtained

$$\begin{aligned} \vec{V}_3 &= \text{sign}(k_3) \mathbf{g}(\overrightarrow{O_1 P_1} \times \overrightarrow{O_1 P_5} \times \overrightarrow{O_1 P_3}) \\ &= -\text{sign}(k_4) \mathbf{g}(\overrightarrow{O_2 P_6} \times \overrightarrow{O_2 P_2} \times \overrightarrow{O_2 P_4}) = \mathbf{g}(\overrightarrow{P_5 P_6}), \end{aligned} \quad (3)$$

where $\text{sign}(s)$ presents the sign function which returns the sign of the variable s and $\mathbf{g}(\vec{S})$ is the function which normalizes the vector \vec{S} and it is defined as

$$\mathbf{g}(\vec{S}) = \begin{cases} \frac{\vec{S}}{\|\vec{S}\|}, & \text{if } \|\vec{S}\| \neq 0, \\ [0, 0, 0], & \text{if } \|\vec{S}\| = 0. \end{cases} \quad (4)$$

TABLE 1: Points for other Dubins curves.

Type	S	C	CS	SC	CSC
Points	P_1, P_2	P_1, P_2	P_1, P_5, P_2	P_1, P_6, P_2	P_1, P_5, P_6, P_2
Constants	/	O_1, P_3	O_1, P_3	O_2, P_4	O_1, P_3, P_4, O_2
Parameters	/	k_1, k_3	k_1, k_3, k_5	k_2, k_4, k_6	k_1, k_2, \dots, k_6

Due to the fact that P_5 and P_6 are the tangent points of C_1 and C_2 with the S segment, there are

$$\begin{aligned} \overrightarrow{P_5 O_1} \times \vec{V}_3 &= 0, \\ \overrightarrow{P_1 O_1} \times \vec{V}_1 &= 0, \end{aligned} \quad (5)$$

$$\overrightarrow{P_6 O_2} \times \vec{V}_3 = 0,$$

$$\overrightarrow{P_2 O_2} \times \vec{V}_2 = 0,$$

$$\overrightarrow{P_5 O_1} = \overrightarrow{P_1 O_1} = R_1, \quad (6)$$

$$\overrightarrow{P_6 O_2} = \overrightarrow{P_2 O_2} = R_2.$$

Furthermore, it can be concluded from (6) that $\overrightarrow{P_3 O_1}$ and $\overrightarrow{P_4 O_2}$ are the angle bisector of $\angle P_3 P_1 O_1$ and $\angle P_4 P_2 O_2$; therefore there are

$$O_1 = P_3 + k_1 \text{sign}(k_3) (-\vec{V}_1 + \vec{V}_3) \quad (7)$$

$$O_2 = P_4 + k_2 \text{sign}(k_4) (-\vec{V}_2 + \vec{V}_3).$$

The parameters k_1 – k_6 can be calculated based on the specified set of equations to determine the CSC curve by substituting the variables in (5) and (6) with corresponding expression in (1)–(3) and (7) if there is no semicircular C segment in the CSC curve. Likewise, other types of Dubins curves, such as S , C , CS , and SC curves, can be determined easily due to the coplanarity of their initial and final configurations and segments. The key points and constants for their determinations are presented in Table 1 and the set of equations for calculation can be obtained as well based on the presented equations above.

If the CSC curve has semicircular C segments, the equations for calculation are built through modifying the corresponding equations above based on the curve's geometric characters. For example, for a CSC curve with semicircular C_1 , the modified equations for its calculation are presented as

$$\vec{V}_3 = -\vec{V}_1, \quad (8)$$

$$\overrightarrow{P_5 O_1} = \overrightarrow{P_1 O_1}.$$

Meanwhile, (2) is substituted by

$$P_6 = P_5 + k_5 \vec{V}_3, \quad (9)$$

where k_5 is constant that indicates the length of the S segment. With these equations there will be sufficient equations to determine the feasible CSC Dubins curve. The other Dubins curves can be determined by a similar technique.

2.3. Dubins Path Planning. Considering the efficiency and optimality in path planning, several candidate Dubins curves are adopted in this paper which are S, C, CS, SC, and CSC in the order of priority from high to low. Meanwhile, these curves can be divided into two classes which are the coplanar curves and the noncoplanar curves according to the coplanarity of the initial and final configurations. The coplanar curves contain the S, C, CS, SC, and few CSC curves because the initial and final configurations of these curves lie in the same plane. The noncoplanar curves consist of the CSC curves with noncoplanar segments. Therefore, the Dubins curve determination approach is expressed as follows. Firstly the coplanarity of the initial and final configurations is evaluated. If they are coplanar, the coplanar curves are selected as the candidate curves and checked in the order of priority. Otherwise the noncoplanar curve is selected and checked instead. Once the feasible curve is obtained, it is adopted as the feasible path. The algorithm of this approach is presented as Algorithm 1.

To simplify the expression of the Dubins curve determination approach, the calculation process of a certain curve is presented as

$$T(Pt, Cf_1, Cf_2, C_r) = \begin{cases} L, & \text{if the curve exists} \\ 0, & \text{otherwise,} \end{cases} \quad (10)$$

where Pt is the desired type of the Dubins curve for determination while Cf_1 and Cf_2 are the initial and final configurations of Pt , respectively, and C_r presents the collection of the feasible radii according to Pt . The function T returns the desired Dubins curve if this curve can be determined; otherwise it returns 0. Besides, the vector \vec{n} is defined to help evaluate the colinearity and coplanarity of the initial and final configurations, where $\vec{n} = \mathbf{g}(\vec{V}_1 \times \vec{V}_2)$.

To prove the validity and efficiency of Algorithm 1, an example is presented and analyzed on the CPU Intel Core I3 6300@3.8 GHz with Matlab. The conditions of this example are set as

$$\begin{aligned} P_1 &= [0, 0, 0], \\ \vec{V}_1 &= [1, 0, 0], \\ P_2 &= [6, 5, 4], \\ \vec{V}_2 &= [-1, -1, -1], \\ R &= 2. \end{aligned} \quad (11)$$

In this example the Dubins curve determination problem is converted into a least squares problem and solved with the Levenberg-Marquardt (L-M) algorithm where $\lambda = 0.005$. The time consumption of Algorithm 1 is 0.683 s with computational accuracy 10^{-5} and the simulation result is illustrated in Figure 2. If the computation resource of the robot is abundant, the evaluation processes of the candidate curves in Algorithm 1 can be implemented at the same time to reduce the time consumption.

Input: $P_1, \vec{V}_1, P_2, \vec{V}_2, R$
Output: feasible Dubins curve L

- (1) **if** $[P_1, \vec{V}_1], [P_2, \vec{V}_2]$ are coplanar, **then**
- (2) **if** $\vec{V}_1 = \vec{V}_2 = \mathbf{g}(\vec{P}_1\vec{P}_2)$, **then**
- (3) $L \leftarrow T(S, [P_1, \vec{V}_1], [P_2, \vec{V}_2])$
- (4) **break**
- (5) **end if**
- (6) **if** $\vec{P}_1\vec{P}_2 = \pm R(\mathbf{g}(\vec{n} \times \vec{V}_1) + \mathbf{g}(\vec{n} \times \vec{V}_2))$, **then**
- (7) $L \leftarrow T(C, [P_1, \vec{V}_1], [P_2, \vec{V}_2], R)$
- (8) **break**
- (9) **end if**
- (10) **if** $\exists P_5, T(C, [P_1, \vec{V}_1], [P_5, \vec{V}_2], R) \neq 0$ and $T(S, [P_5, \vec{V}_2], [P_2, \vec{V}_2]) \neq 0$, **then**
- (11) $L \leftarrow T(CS, [P_1, \vec{V}_1], [P_2, \vec{V}_2], R)$
- (12) **break**
- (13) **end if**
- (14) **if** $\exists P_6, T(S, [P_1, \vec{V}_1], [P_6, \vec{V}_1]) \neq 0$ and $T(C, [P_6, \vec{V}_1], [P_2, \vec{V}_2], R) \neq 0$, **then**
- (15) $L \leftarrow T(SC, [P_1, \vec{V}_1], [P_2, \vec{V}_2], R)$
- (16) **break**
- (17) **end if**
- (18) $L \leftarrow T(CSC, [P_1, \vec{V}_1], [P_2, \vec{V}_2], [R, R])$
- (19) **break**
- (20) **else**
- (21) $L \leftarrow T(CSC, [P_1, \vec{V}_1], [P_2, \vec{V}_2], [R, R])$
- (22) **break**
- (23) **end if**

ALGORITHM 1: Dubins curve determination algorithm.

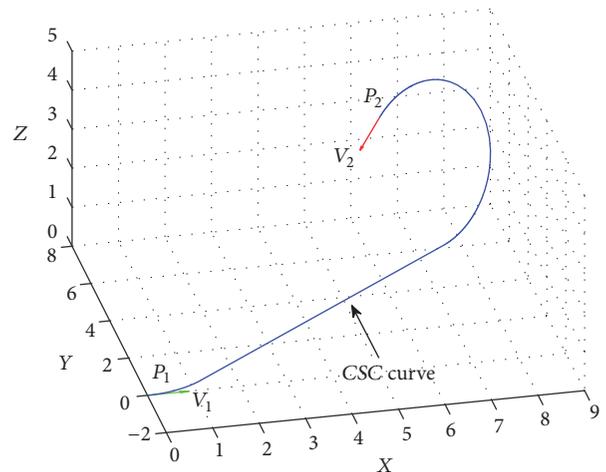


FIGURE 2: The result of Algorithm 1 in Dubins curve determination.

3. Dubins-APF Path Planning

Although the Dubins curves are feasible to form the path for docking with posture constraints, it is difficult to determine the collision-free path and ensure the safety of the robot only using Algorithm 1 when there are unknown obstacles

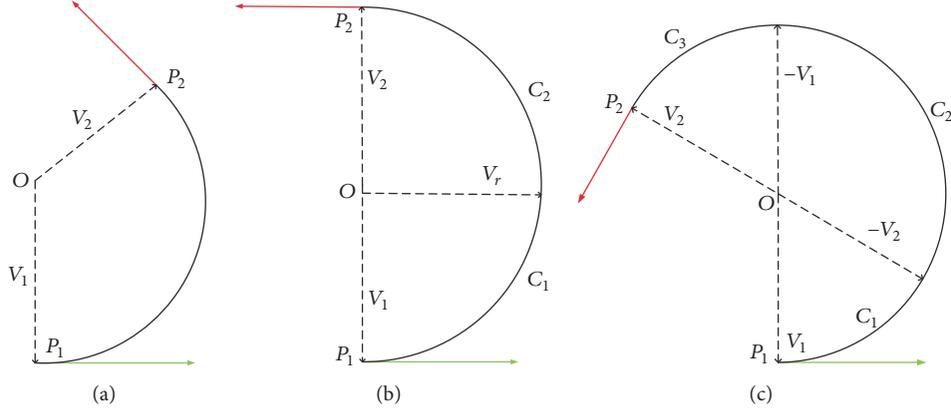


FIGURE 3: Different types of C segments. (a) The minor C segment; (b) the semicircular C segment; (c) the major C segment.

in the workspace. To overcome this shortcoming, a path planning strategy is designed and introduced in this section via combining the Dubins curves with the APF approach to avoid collision with obstacles.

In this paper both static obstacles and moving obstacles are considered. The position and velocity of the obstacle i are written as O_i and \vec{V}_{O_i} separately. Assuming that the minimal safe distance between the robot and obstacle is δ_s , the radius of the obstacle i can be written as $R'_i = R_i + \delta_s$, where R_i is the physical radius of the obstacle.

3.1. Collision Prediction. Collision prediction is one of the fundamental issues of the obstacle avoidance researches [34, 35]. If the planned path is written as L , the motion of the robot following L is presented as $P_r(t)$, where $t \in [0, t_f]$ and t_f is the final time the robot follows L . This issue is easy to address in this paper because the planned path solely consists of the Dubins curves. Therefore once L is obtained, the parameters of these Dubins curves can be utilized to evaluate the feasibility of the planned path in collision prediction. First of all, $\Theta(\vec{V}_i, \vec{V}_j)$ is defined as the operator which calculates and returns the angle between the initial vector \vec{V}_i and the final vector \vec{V}_j .

For following the S segment $\overline{P_1 P_2}$, the movement of the robot on it is simple and can be described as

$$P_r(t) = P_1 + tV_1, \quad (12)$$

where $t \in [0, \|\overline{P_1 P_2}\|/v]$.

There are three types of C segments that one Dubins curve might have which are the minor C segment, the semicircular C segment, and the major C segment. These C segments are illustrated in Figure 3, respectively.

The illustration of the minor C segment is presented in Figure 3(a). For following the minor C segment $\widehat{P_1 P_2}$, $P_r(t)$ is expressed as

$$P_r(t) = O + \frac{\sin(\theta - t(v/R))\vec{V}_1 + \sin(t(v/R))\vec{V}_2}{\sin(\theta)}, \quad (13)$$

TABLE 2: Parameters of c_1 , c_2 , and c_3 .

Index	\vec{V}_i and \vec{V}_j	Central angle θ	Initial and final time
c_1	$\vec{V}_1, -\vec{V}_2$	$\Theta(\vec{V}_1, -\vec{V}_2)$	$0, \theta_1 R/v$
c_2	$-\vec{V}_2, -\vec{V}_1$	$\Theta(-\vec{V}_2, -\vec{V}_1)$	$t_1, t_1 + \theta_2 R/v$
c_3	$-\vec{V}_1, \vec{V}_2$	$\Theta(-\vec{V}_1, \vec{V}_2)$	$t_2, t_2 + \theta_3 R/v$

where $t \in [0, (R/v)\Theta(\overrightarrow{OP_1}, \overrightarrow{OP_2})]$, $\vec{V}_1 = \mathbf{g}(\overrightarrow{OP_1})$, $\vec{V}_2 = \mathbf{g}(\overrightarrow{OP_2})$, and $\theta = \Theta(\vec{V}_1, \vec{V}_2)$.

Similarly, the major C segment is shown in Figure 3(c). $Pr(t)$ can be determined by dividing the major C segment into three minor C segments which are named c_1 , c_2 , and c_3 . If the initial vector and the final vector of the major C segment are expressed as $\vec{V}_1 = \mathbf{g}(\overrightarrow{OP_1})$ and $\vec{V}_2 = \mathbf{g}(\overrightarrow{OP_2})$, the parameters of c_1 , c_2 , and c_3 are concluded in Table 2. Meanwhile, the movement on each minor C segment can be determined according to (13) and $Pr(t)$ of the major C segment can be obtained by their combination.

The semicircular C segment is shown in Figure 3(b). Likewise, it can be divided into two minor C segments which are c_1 and c_2 . The initial and final vectors of c_1 and c_2 are \vec{V}_1 , $\vec{V}_r(0)$ and $\vec{V}_r(0)$, \vec{V}_2 , respectively, where $\vec{V}_r(0)$ is the initial pose of the robot following the semicircular C segment. The movements on c_1 and c_2 can be determined according to (13) and $Pr(t)$ can be obtained by their combination.

Based on $P_r(t)$ of each C segment and S segment of the planned path, the movement of the robot following certain Dubins curve can be specified as well. The feasibility of L can be determined by evaluating these path segments in the temporal order. If there are $\|P_r(t) - O_i(t)\| \geq R'_i$ for all the obstacles, L is collision-free. If L is not collision-free as Figure 4 shows, the time when $\|P_r(t) - O_i(t)\|$ is minimal is defined as the maximal collision time T_{C_i} for O_i . At T_{C_i} the distance between O_i and P_r is defined as the minimal collision distance D_{C_i} for O_i and $D_{C_i} = \|P_r(T_{C_i}) - O_i(T_{C_i})\|$. Therefore if $D_{C_i} < R'_i$, the robot will collide with the obstacle O_i by following L and new L is needed.

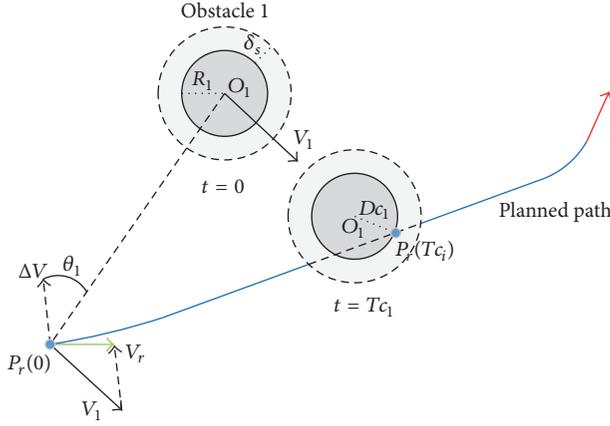


FIGURE 4: The planned path is not collision-free.

3.2. Path Replanning Strategy. Once the planned path is not collision-free, new path is required immediately to avoid collision for the safety of robot. However, sometimes only local knowledge about the environment can be obtained in practice while the replanning process could not be implemented in advance [36–39]. Moreover, if the area for docking is crowded with obstacles, the efficiency of the path replanning process is critical.

To solve this problem, a reactive path replanning strategy is proposed in this paper. The conception of this replanning strategy is to construct the obstacle avoidance potential field to determine the feasible intermediate configuration and the pass-by path is generated to avoid the obstacle based on this intermediate configuration. Then the follow-up path from the intermediate configuration to the docking station is generated to complete the path for docking. The details of this proposed replanning strategy are described as follows.

3.2.1. Obstacle Avoidance Potential Field Design. The obstacle avoidance potential field is proposed in this paper to help determine the intermediate configuration for path replanning inspired by the methods proposed in [30, 31, 40]. The obstacle avoidance potential field is a compound repulsive potential field which consists of two components U_p and U_v . These components are caused by the positions and velocities of the obstacles, respectively. For the obstacle O_i , its corresponding repulsive potential field U_{p_i} is designed as

$$U_{p_i}(\vec{P}_r O_i) = \begin{cases} k_s e^{-\|\vec{P}_r O_i\|}, & \|\vec{P}_r O_i\| < \|\vec{P}_r P_d\| \\ 0, & \text{otherwise,} \end{cases} \quad (14)$$

where k_s is a constant. The corresponding repulsive potential field force \vec{F}_{p_i} is designed as

$$\vec{F}_{p_i}(\vec{P}_r O_i) = -\nabla U_{p_i} = -k_s e^{-\|\vec{P}_r O_i\|} \mathbf{g}(\vec{P}_r O_i), \quad (15)$$

when $\|\vec{P}_r O_i\| < \|\vec{P}_r P_d\|$.

If the obstacle O_i is a moving obstacle, the velocity difference between it and the robot is expressed as $\delta\vec{V} = \vec{V}_r -$

\vec{V}_{O_i} . If \vec{H}_i is defined that $\vec{H}_i = \mathbf{g}(\vec{P}_r O_i \times \delta\vec{V} \times \vec{P}_r O_i)$, the corresponding repulsive potential field U_{v_i} is expressed as

$$U_{v_i}(\vec{H}_i) = \begin{cases} -k_v \|\delta\vec{V}\| e^{-\|\vec{P}_r O_i\|}, & \|\vec{P}_r O_i\| < \|\vec{P}_r P_d\| \\ 0, & \text{otherwise,} \end{cases} \quad (16)$$

where $\theta_i = \Theta(\vec{V}_r, \vec{P}_r O_i)$ and k_v is a constant. The corresponding repulsive force is expressed as

$$\vec{F}_{v_i} = -\nabla U_{v_i} = k_v \|\delta\vec{V}\| e^{-\|\vec{P}_r O_i\|} \vec{H}_i, \quad (17)$$

when $\|\vec{P}_r O_i\| < \|\vec{P}_r P_d\|$.

The total repulsive field force caused by O_i can be derived from F_{p_i} and F_{v_i} as

$$\vec{F}_r = \sum_{i=1}^N (-\nabla U_{p_i} - \nabla U_{v_i}) = \sum_{i=1}^N (\vec{F}_{p_i} + \vec{F}_{v_i}), \quad (18)$$

where N is the number of the detected obstacles that $\|\vec{P}_r O_i\| < \|\vec{P}_r P_d\|$.

In this paper \vec{F}_r is used to determine the feasible intermediate configuration for obstacle avoidance while considering both the positions and velocities of the obstacles.

3.2.2. Intermediate Configuration Determination. In this paper the feasible intermediate configuration is utilized to generate collision-free path according to \vec{F}_r . The intermediate configuration $[P_m, \vec{V}_m]$ is determined as follows. Firstly, the obstacle O_g is defined as the target obstacle O_g for determining the intermediate configuration, where

$$O_g = \{O_g \mid T_{c_g} = \min\{T_{c_1}, T_{c_2}, \dots, T_{c_i}, \dots, T_{c_N}\}\}. \quad (19)$$

Then the final pose of the path segment that $P_r(T_{c_g})$ is on is chosen as the intermediate pose. Therefore, the original intermediate point P_m is expressed as

$$P_m = O_g(T_{c_g}) + (R'_g + \alpha) \mathbf{g}(\vec{F}_m), \quad (20)$$

where α is the parameter to adjust the position of the intermediate point. The larger α is, the more conservative the path replanning strategy is. \vec{F}_m is defined as the intermediate direction where

$$\vec{F}_m = \begin{cases} \mathbf{g}(\vec{V}_m \times \vec{F}_r \times \vec{V}_m), & \vec{V}_m \times \vec{F}_r \neq 0 \\ \mathbf{g}(\vec{V}_m \times \vec{F}_{\text{rand}} \times \vec{V}_m), & \text{otherwise,} \end{cases} \quad (21)$$

where \vec{F}_{rand} is a nonzero random vector and $\mathbf{g}(\vec{F}_{\text{rand}}) \neq \mathbf{g}(\vec{F}_r)$. At last the Dubins curve L' from $[P_r, \vec{V}_r]$ to $[P_m, \vec{V}_m]$ with radii R_t and R_g is generated as the pass-by path L' .

If L' is collision-free, the intermediate configuration is adopted and L' is adopted as the pass-by path $L_{\text{pass-by}}$. If L' is not collision-free, the new target obstacle can be determined and the new intermediate point P'_m can be obtained as well. The pose of the new intermediate vector is chosen as $\vec{V}'_m = \vec{V}_m$ and the intermediate configuration determination process is repeated.

Input: $[P_r, \vec{V}_r], [P_d, \vec{V}_d], R_t, [O, \vec{V}_O, R'], L, R_1, R_2, k_s, k_v, \alpha$

Output: L

- (1) $\text{rpflag} = 0$
- (2) **while true do**
- (3) $T_{\min} = \min\{T_{C_i} \mid DC_i < R'_i, i \in [0, t_f]\}$
- (4) **if** $T_{\min} > 0$ **then**
- (5) $T_g \leftarrow T_{\min}$, obtain $O_g, R'_g, P_m, \vec{V}_m$
- (6) **if** $\text{rpflag} == 0$ **then**
- (7) $R_1 \leftarrow R_2, R_2 \leftarrow R'_g$
- (8) **end if**
- (9) $L_{\text{pass-by}} \leftarrow T([P_r, \vec{V}_r], [P_m, \vec{V}_m], [R_1, R_2])$
- (10) **if** $L_{\text{pass-by}}$ is not collision-free **then**
- (11) $L \leftarrow L_{\text{pass-by}}, \text{rpflag} = 1$
- (12) **continue**
- (13) **end if**
- (14) $R_1 \leftarrow R_2, R_2 \leftarrow R_t$
- (15) $L_{\text{follow-up}} \leftarrow T([P_m, \vec{V}_m], [P_d, \vec{V}_d], [R_1, R_2])$
- (16) **if** $L_{\text{follow-up}}$ is not collision-free **then**
- (17) Replan $L_{\text{follow-up}}$ with Algorithm 2
- (18) **else**
- (19) $L \leftarrow L_{\text{pass-by}} + L_{\text{follow-up}}$
- (20) **end if**
- (21) **else**
- (22) **break**
- (23) **end if**
- (24) **end while**

ALGORITHM 2: Path replanning algorithm.

3.2.3. Path Generation. The replanned path L consists of two subpaths which are the pass-by path $L_{\text{pass-by}}$ and the follow-up path $L_{\text{follow-up}}$. The initial and final configurations of $L_{\text{pass-by}}$ are $[P_r, \vec{V}_r]$ and $[P_m, \vec{V}_m]$, respectively, while the initial and final configurations of $L_{\text{follow-up}}$ are $[P_m, \vec{V}_m]$ and $[P_d, \vec{V}_d]$. Since $L_{\text{pass-by}}$ is obtained in the intermediate configuration determination process, $L_{\text{follow-up}}$ is generated after $L_{\text{pass-by}}$ is determined. If $L_{\text{follow-up}}$ is collision-free, the combination of $L_{\text{pass-by}}$ and $L_{\text{follow-up}}$ is set as the replanned path for docking and followed till this path is evaluated to be infeasible. Otherwise, $L_{\text{follow-up}}$ is replanned according to the replanning strategy as well.

This proposed path replanning strategy is concluded in Algorithm 2. In Algorithm 2 rpflag is the flag bit which indicates whether a feasible path has been planned ($\text{rpflag} == 1$) or not ($\text{rpflag} == 0$). Once Algorithm 2 is called, the minimal maximal collision time T_{\min} is calculated firstly to evaluate whether a new path should be replanned. If $T_{\min} > 0$, it means the new path is required and the path replanning strategy is implemented. If $T_{\min} \leq 0$, it means the planned path L is collision-free and the replanned path is accepted as the feasible path for docking.

The parameters of Algorithm 2 are selected according the following rules. The obstacle avoidance strategy is affected by k_s/k_v . The larger k_s/k_v is, the more attentions are paid on

avoiding the static obstacles and vice versa. Therefore if the moving obstacles in the environment for docking are large and fast, a small k_s/k_v is preferred to make the path planner sensitive to the moving obstacles. α provides the margin for robot controller in handling the emergency situations such as the velocity change of the moving obstacle and ocean currents. The larger α is, the more potentials are preserved for robot control but, however, the longer the planned path may be. Therefore α is chosen based on the estimation of the environment for docking.

3.3. DAPF Path Planning Approach. The DAPF path planning approach can be concluded as Algorithm 3 based on the Dubins curves and the path replanning strategy. In Algorithm 3 the configuration of the known obstacles is written as \mathcal{C} . The DAPF path planning approach works in a reactive way to the change of the environment as follows. Firstly, an initial path is generated as the target path L and followed by the robot. Then the feasibility of L is evaluated when \mathcal{C} is changed. If L is infeasible, then the new path is generated and L is substituted by the new path. Otherwise \mathcal{C} is updated and L is followed.

An example of the DAPF path planning approach is given in Figure 5, where

$$\begin{aligned}
 P_r &= [0, 0, 0], \\
 \vec{V}_r &= [1, 0, 0], \\
 P_d &= [6, 5, 4], \\
 \vec{V}_d &= [1, -0, 0], \\
 R_t &= 2, \\
 R_s &= 6, \\
 \alpha &= 0.1, \\
 O_1 &= [9, 0, -4], \\
 \vec{V}_{O_1} &= [0, 0, 0.5], \\
 R'_1 &= 3.
 \end{aligned} \tag{22}$$

The result of this scenario shows that the proposed path planning strategy is feasible in generating collision-free path.

4. Simulation and Discussion

In this section simulation experiments are given to prove the validity of the proposed DAPF approach in solving the path planning problem for docking in unknown environment. Meanwhile, the traditional APF approach is adopted in these scenarios to compare the path planning performance with the

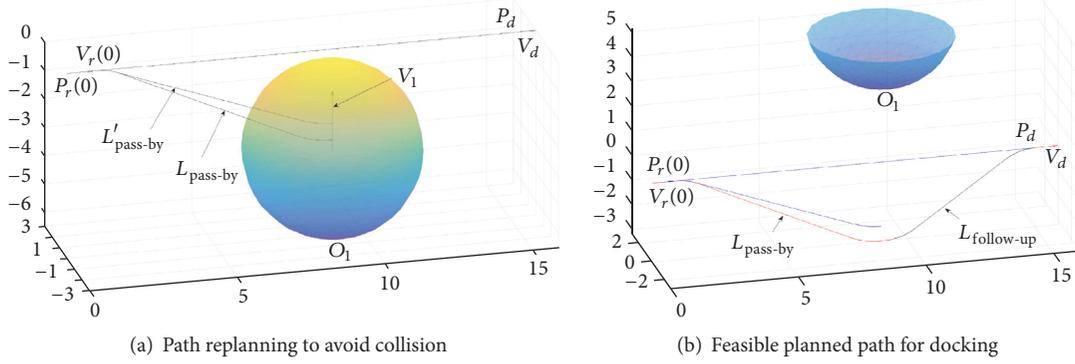


FIGURE 5: Example of the DAPF path planning strategy in environment with one obstacle. (a) The intermediate configuration is determined as well as $L_{\text{pass-by}}$. (b) The planned path is collision-free and feasible for docking.

Input: $[P_r, \vec{V}_r], [P_d, \vec{V}_d], R_t, k_s, k_v, \alpha$

Output: L

- (1) $L \leftarrow T([P_r, \vec{V}_r], [P_d, \vec{V}_d], [R_t, R_t])$
- (2) $R_1 \leftarrow R_t, R_2 \leftarrow R_t$
- (3) $\mathcal{C}' \leftarrow [O, \vec{V}_0, R']$
- (4) **while** $[P_r, \vec{V}_r] \neq [P_d, \vec{V}_d]$ **do**
- (5) **if** $\mathcal{C}' \notin \mathcal{C}$ **then**
- (6) $L \leftarrow \text{Algorithm 2}$
- (7) **end if**
- (8) $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{C}'$
- (9) follow L
- (10) **end while**

ALGORITHM 3: DAPF path planning algorithm.

proposed path planning approach. The common parameters of these scenarios are set as

$$\begin{aligned}
 P_r(0) &= [0, 0, 0], \\
 \vec{V}_r(0) &= [1, 0, 0], \\
 P_d &= [15, 0, 0], \\
 \vec{V}_d &= [1, 0, 0], \\
 R_t &= 2, \\
 R_s &= 6,
 \end{aligned} \tag{23}$$

and the step length of these simulations is set as 0.5. The traditional APF approach is designed as

$$\begin{aligned}
 U_{\text{attr}}(\vec{P}_r O_i) &= \frac{k_a}{2} \left(\frac{1}{\|\vec{P}_r O_i\| - R_i} - \frac{1}{\|\vec{P}_r P_d\|} \right)^2, \\
 U_{\text{repu}}(\vec{P}_r O_i) &= k_r \left(\frac{1}{\|\vec{P}_r O_i\| - R_i} \right),
 \end{aligned} \tag{24}$$

where $k_a = 1$ and $k_r = 5$. The parameters of the DAPF approach are $k_s = 1, k_v = 1$, and $\alpha = 0.1$.

4.1. Scenario 1. Both static and moving obstacles are considered in scenario 1. In this scenario, one of the static obstacles is set close to the docking station which simulates a goal nonreachable problem identified in [41, 42]. The configurations of the obstacles are set as

$$\begin{aligned}
 O_1 &= [4, 2, 1], \\
 \vec{V}_1 &= [0, 0, 0], \\
 R'_1 &= 1, \\
 O_2 &= [13, -2, 0], \\
 \vec{V}_2 &= [0, 0, 0], \\
 R'_2 &= 2, \\
 O_3 &= [7, 0, -4], \\
 \vec{V}_3 &= [0, 0, 0.5], \\
 R'_3 &= 3.
 \end{aligned} \tag{25}$$

The path planning results of the traditional APF and DAPF approaches are illustrated in Figure 6 and Table 3. From Figure 6 it can be concluded that the path planned by the APF approach (cyan cross line) is affected by the obstacle near the docking station and the final configuration of its planned path is infeasible for docking. The path planned by the DAPF (black solid line) is collision-free and its final configuration is $V_r = [0.9934, 0.1097, 0.0331]$. It is also proved in this scenario that the DAPF approach is capable of planning the feasible path even if the environment near the docking station is complex.

TABLE 3: Results of scenario 1.

Approach	Final pose	Reach time	$\ \Delta\vec{V}\ $
APF	[0.0051, 0.0564, -0.9984]	21.0000	1.4106
DAPF	[0.9934, 0.1097, 0.0331]	24.5102	0.1148

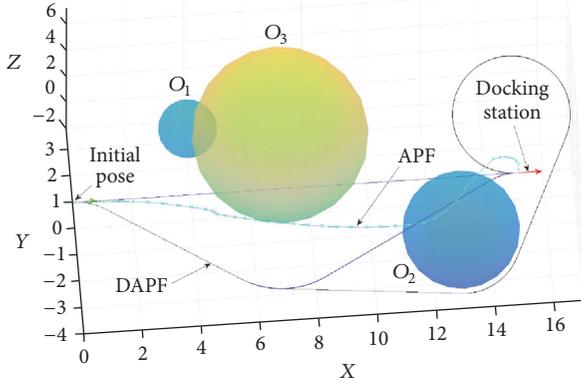


FIGURE 6: Scenario 1: path planning for docking in the environment with static and moving obstacles.

4.2. *Scenario 2.* In scenario 2, the local minima problem is considered and modeled as Figure 7 shows, where the configurations of the obstacles are set as

$$\begin{aligned}
 O_1 &= [7, -2, -2], \\
 O_2 &= [7, -2, 2], \\
 O_3 &= [7, 2, 2], \\
 O_4 &= [7, 2, -2], \\
 O_5 &= [7, -2\sqrt{2}, 0], \\
 O_6 &= [7, 0, 2\sqrt{2}], \\
 O_7 &= [7, 0, -2\sqrt{2}], \\
 O_8 &= [7, 2\sqrt{2}, 0], \\
 O_9 &= [9, 0, 0], \\
 R_1 &= 2, \\
 R_2 &= 2, \\
 R_3 &= 2, \\
 R_4 &= 2, \\
 R_5 &= 3,
 \end{aligned} \tag{26}$$

and the velocities of these obstacles are all set as $[0, 0, 0]$. Therefore, these obstacles form a compound concave obstacle together which will cause the local minima problem in the traditional APF approach.

As Figure 7 and Table 4 show, the path planned by the APF approach (cyan cross line) encounters the local minima

TABLE 4: Results of scenario 2.

Approach	Final pose	Reach time	$\ \Delta\vec{V}\ $
APF	/	∞	/
DAPF	[0.9999, 0.0000, 0.0001]	17.9533	0.0001

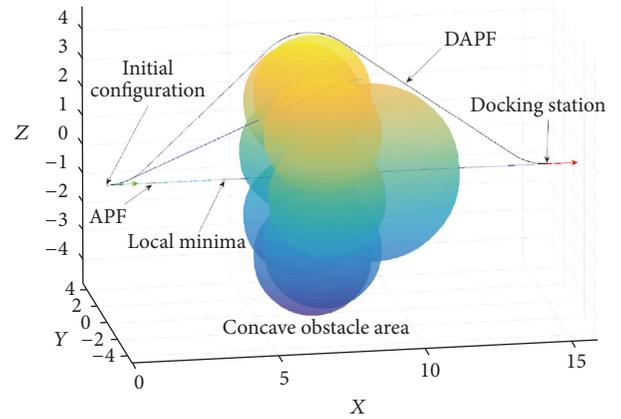


FIGURE 7: Scenario 2: path planning for docking in environment with concave obstacle.

problem and gets stuck at the local minima area while the path planned by the DAPF approach (black solid line) is free from the affection of the local minima and reaches the docking station with feasible pose in spite of the fact that a modified APF is utilized in path replanning.

5. Conclusion and Future Work

In this paper a path planning approach named DAPF is presented for autonomous robot docking based on the combination of the Dubins curves and the artificial potential field approach. Firstly, the determination approach of the Dubins curves is proposed and the collision prediction approach of the planned path is realized based on this approach. Then the path replanning strategy is proposed with the help of the obstacle avoidance potential field. In the replanning strategy the intermediate configuration is calculated based on the obstacle avoidance potential field and adopted to determine the new path. The path planning task is completed through implementing the planning and replanning process in a reactive mode to the changes of environment. Simulation results are also presented to prove the feasibility of the DAPF approach through comparison with the traditional APF approach.

In this proposed approach, little prior knowledge about the environment is required and the planned path is feasible for autonomous robot to follow. As a kind of reactive path planning approaches, this approach is easy to execute by the CPU of the robot in practice. Better performance can be achieved in the future work via adapting parallel computation techniques to enhance the real-time capability of this approach. Specific models of the autonomous robots

will be considered to improve the feasibility of this approach in practical applications such as underwater docking and parking.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant nos. 51579210, 61472325, 61472326, 61733014, and 61633002.

References

- [1] D. Floreano and R. J. Wood, "Science, technology and the future of small autonomous drones," *Nature*, vol. 521, no. 7553, pp. 460–466, 2015.
- [2] Z. Li, S. Zhao, J. Duan, C. Su, C. Yang, and X. Zhao, "Human cooperative wheelchair with brain machine interaction based on shared control strategy," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 1, pp. 185–195, 2017.
- [3] C. Chin and M. Lau, "Modeling and testing of hydrodynamic damping model for a complex-shaped remotely-operated vehicle for control," *Journal of Marine Science and Application*, vol. 11, no. 2, pp. 150–163, 2012.
- [4] R. Cui, C. Yang, Y. Li, and S. Sharma, "Adaptive Neural Network Control of AUVs With Control Input Nonlinearities Using Reinforcement Learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 6, pp. 1019–1029, 2017.
- [5] J. Petit and S. E. Shladover, "Potential Cyberattacks on Automated Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 546–556, 2015.
- [6] C. S. Chin and S. H. Lum, "Rapid modeling and control systems prototyping of a marine robotic vehicle with model uncertainties using xPC Target system," *Ocean Engineering*, vol. 38, no. 17–18, pp. 2128–2141, 2011.
- [7] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich, "A review of space robotics technologies for on-orbit servicing," *Progress in Aerospace Sciences*, vol. 68, pp. 1–26, 2014.
- [8] C. W. Bac, E. J. Van Henten, J. Hemming, and Y. Edan, "Harvesting Robots for High-value Crops: State-of-the-art Review and Challenges Ahead," *Journal of Field Robotics*, vol. 31, no. 6, pp. 888–911, 2014.
- [9] D. Verstraete, A. Gong, D. D.-C. Lu, and J. L. Palmer, "Experimental investigation of the role of the battery in the AeroStack hybrid, fuel-cell-based propulsion system for small unmanned aircraft systems," *International Journal of Hydrogen Energy*, vol. 40, no. 3, pp. 1598–1606, 2015.
- [10] L. Wu, Y. Li, S. Su, P. Yan, and Y. Qin, "Hydrodynamic analysis of AUV underwater docking with a cone-shaped dock under ocean currents," *Ocean Engineering*, vol. 85, pp. 110–126, 2014.
- [11] N. Mathew, S. L. Smith, and S. L. Waslander, "Multirobot rendezvous planning for recharging in persistent tasks," *IEEE Transactions on Robotics*, vol. 31, no. 1, pp. 128–142, 2015.
- [12] Z. Yan, C. Deng, D. Chi, T. Chen, and S. Hou, "Path planning method for UUV homing and docking in movement disorders environment," *Scientific World Journal*, vol. 2014, Article ID 246469, 2014.
- [13] K. Teo, E. An, and P.-P. J. Beaujean, "A robust fuzzy autonomous underwater vehicle (AUV) docking approach for unknown current disturbances," *IEEE Journal of Oceanic Engineering*, vol. 37, no. 2, pp. 143–155, 2012.
- [14] D. Herrero, J. Villagr, and H. Martínez, "Self-configuration of waypoints for docking maneuvers of flexible automated guided vehicles," *IEEE Transactions on Automation Science Engineering*, vol. 10, no. 2, pp. 470–475, 2013.
- [15] C. Yang, Z. Li, R. Cui, and B. Xu, "Neural network-based motion control of an underactuated wheeled inverted pendulum model," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 11, pp. 2004–2016, 2014.
- [16] J. Zhang, S. Zhao, Y. Zhang, and Y. Li, "Optimal planning approaches with multiple impulses for rendezvous based on hybrid genetic algorithm and control method," *Advances in Mechanical Engineering*, vol. 7, no. 3, pp. 1–11, 2015.
- [17] D. Zhu, W. Li, M. Yan, and S. X. Yang, "The path planning of AUV based on D-S information fusion map building and bio-inspired neural network in unknown dynamic environment," *International Journal of Advanced Robotic Systems*, vol. 11, no. 1, pp. 415–429, 2014.
- [18] O. Motlagh, D. Nakhaeinia, S. H. Tang, B. Karasfi, and W. Khaksar, "Automatic navigation of mobile robots in unknown environments," *Neural Computing and Applications*, vol. 24, no. 7–8, pp. 1569–1581, 2014.
- [19] R. Cui, Y. Li, and W. Yan, "Mutual information-based multi-AUV path planning for scalar field sampling using multidimensional RRT*," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 7, pp. 993–1004, 2016.
- [20] B. Kovács, G. Szayer, F. Tajti, M. Burdelis, and P. Korondi, "A novel potential field method for path planning of mobile robots by adapting animal motion attributes," *Robotics and Autonomous Systems*, vol. 82, pp. 24–34, 2016.
- [21] Q. Huang and G. Zheng, "Route Optimization for Autonomous Container Truck Based on Rolling Window," *International Journal of Advanced Robotic Systems*, vol. 13, no. 3, Article ID 64116, 2016.
- [22] Á. V. F. M. De Oliveira and M. A. C. Fernandes, "Dynamic planning navigation strategy for mobile terrestrial robots," *Robotica*, vol. 34, no. 3, pp. 568–583, 2016.
- [23] M. Hoy, A. S. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: A survey," *Robotica*, vol. 33, no. 3, pp. 463–497, 2015.
- [24] A. M. Shkel and V. Lumelsky, "Classification of the Dubins set," *Robotics and Autonomous Systems*, vol. 34, no. 4, pp. 179–202, 2001.
- [25] R. G. Sanfelice, S. Z. Yong, and E. Frazzoli, "On minimum-time paths of bounded curvature with position-dependent constraints," *Automatica. A Journal of IFAC, the International Federation of Automatic Control*, vol. 50, no. 2, pp. 537–546, 2014.
- [26] M. Shanmugavel, A. Tsoordos, B. White, and R. Zbikowski, "Co-operative path planning of multiple UAVs using Dubins paths with clothoid arcs," *Control Engineering Practice*, vol. 18, no. 9, pp. 1084–1092, 2010.
- [27] Y. Lin and S. Saripalli, "Path planning using 3D dubins curve for unmanned aerial vehicles," in *Proceedings of the 2014 International Conference on Unmanned Aircraft Systems, ICUAS 2014*, pp. 296–304, Orlando, FL, USA, May 2014.
- [28] O. Cetin, I. Zagli, and G. Yilmaz, "Establishing obstacle and collision free communication relay for UAVs with artificial

- potential fields,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 69, no. 1-4, pp. 361–372, 2013.
- [29] A. Melingui, R. Merzouki, J. B. Mbede, and T. Chettibi, “A novel approach to integrate artificial potential field and fuzzy logic into a common framework for robots autonomous navigation,” *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering*, vol. 228, no. 10, pp. 787–801, 2014.
- [30] O. Montiel, R. Sep, U. Orozco-Rosas, and R. Sepúlveda, “Optimal path planning generation for mobile robots using parallel evolutionary artificial potential field,” *Journal of Intelligent Robotic Systems*, vol. 79, no. 2, p. 21, 2015.
- [31] H. Min, Y. Lin, S. Wang, F. Wu, and X. Shen, “Path planning of mobile robot by mixing experience with modified artificial potential field method,” *Advances in Mechanical Engineering*, vol. 7, no. 11, Article ID 1687814015619276, 2015.
- [32] M. Shanmugavel, A. Tsourdos, R. Zbikowski, and B. A. White, “3D dubins sets based coordinated path planning for swarm of UAVs,” in *Proceedings of the AIAA Guidance, Navigation, and Control Conference 2006*, pp. 1418–1437, August 2006.
- [33] Y. Wang, S. Wang, M. Tan, C. Zhou, and Q. Wei, “Real-time dynamic Dubins-Helix method for 3-D trajectory smoothing,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 730–736, 2015.
- [34] T. M. Howard, “Model-predictive motion planning: Several key developments for autonomous mobile robots,” *IEEE Robotics Automation Magazine*, vol. 21, no. 21, pp. 64–73, 2014.
- [35] Y. Lu, Z. Xi, and J.-M. Lien, “Online collision prediction among 2D polygonal and articulated obstacles,” *International Journal of Robotics Research*, vol. 35, no. 5, pp. 476–500, 2015.
- [36] P. Mobadersany, S. Khanmohammadi, and S. Ghaemi, “A fuzzy multi-stage path-planning method for a robot in a dynamic environment with unknown moving obstacles,” *Robotica*, vol. 33, no. 9, pp. 1869–1885, 2015.
- [37] J. Peng, W. Luo, W. Liu, W. Yu, and J. Wang, “A suboptimal and analytical solution to mobile robot trajectory generation amidst moving obstacles,” *Autonomous Robots*, vol. 39, no. 1, 2015.
- [38] M. Otte and E. Frazzoli, “RRTX: Asymptotically optimal single-query sampling-based motion planning with quick replanning,” *International Journal of Robotics Research*, vol. 35, no. 7, pp. 797–822, 2016.
- [39] B. H. Lee, J. D. Jeon, and J. H. Oh, “Velocity obstacle based local collision avoidance for a holonomic elliptic robot,” *Autonomous Robots*, pp. 1–17, 2016.
- [40] Y.-b. Chen, G.-c. Luo, Y.-s. Mei, J.-q. Yu, and X.-l. Su, “UAV path planning using artificial potential field method updated by optimal control theory,” *International Journal of Systems Science. Principles and Applications of Systems and Integration*, vol. 47, no. 6, pp. 1407–1420, 2016.
- [41] S. S. Ge and Y. J. Cui, “New potential functions for mobile robot path planning,” *IEEE Transactions on Robotics and Automation*, vol. 16, no. 5, pp. 615–620, 2000.
- [42] U. Orozco-Rosas, O. Montiel, and R. Sepúlveda, “Pseudo-bacterial potential field based path planner for autonomous mobile robot navigation,” *International Journal of Advanced Robotic Systems*, vol. 12, 2015.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

