Path Generation of Autonomous Approach to a Moving Ship for Unmanned Vehicles

Yu Wang, Shuo Wang, and Min Tan

***Abstract*—This paper is devoted to an autonomous approach to a moving ship for unmanned vehicles. The problem is introduced and formulated first. Cooperation between the vehicle and the ship is needed, such that the moving ship to be approached is set to move at a constant speed along a straight line or a circle. Hence, an approaching algorithm based on 3-D path generation is proposed. The 3-D vehicle trajectory produced by the path generation realizes smooth transition from the initial pose to the final pose with geometric constraints. The final poses the unmanned vehicle and the ship's reach simultaneously are analyzed and obtained based on 3-D path generation. The related parameters for autonomous navigation of an unmanned vehicle are discussed according to the straight line or the circle that the ship sails along. The approaching algorithm is sequentially summarized. The simulations are conducted, and the results show the effectiveness and real-time performance of the proposed method.**

***Index Terms*—Autonomous approach, path generation, trajectory smoothing, unmanned vehicles.**

I. Introduction

Unmanned vehicles are increasingly used in civilian and military applications such as rescue, search, data collection, and scouting [1]–[5]. Many control systems and algorithms are developed for unmanned systems to fulfill diversified missions. One of the control system requirements is autonomy, i.e., the capability of detecting internal and external changes and reacting to them without human intervention in a safe and efficient manner [6]. Lin *et al.* proposed autonomous target following using the embedded vision system for an unmanned rotorcraft [7]. Zhang *et al.* designed autonomous task allocation for heterogeneous unmanned vehicles [8], [9]. Likewise, autonomy is essential for the navigation of unmanned underwater or air vehicles to a moving base as well. After an unmanned vehicle completes tasks, it should return to a moving platform such as a ship or a truck automatically. Thus, this paper mainly concentrates on the problem that unmanned underwater or air vehicles achieve autonomous approach to a moving mother ship.

Recently, a few methods have been applied for the autonomous navigation of unmanned vehicles. Ambrosino *et al.* developed an approaching path for unmanned aerial vehicles, which can connect an initial pose to a fixed final pose belonging to the landing track [10]. Storvik *et al.* employed B-splines curves to achieve autonomous approach to a stationary ship. However, it may be relatively difficult to achieve autonomous navigation to a moving ship [11]. Drawing from [10] and [11], we apply path generation to produce a feasible trajectory, which can guide an unmanned vehicle to approach a moving ship. Path generation is aimed at finding a feasible path connecting an initial point to a final point in terms of path length, fuel consumption, or obstacle avoidance. Many numerical algorithms have been applied for path generation [12]. The rapidly exploring random tree methods were employed for path planning [13]–[15]. The A^* algorithms were also used for path generation [16], [17]. Moreover, the mixed-integer linear programming method was utilized to find the path [18]. The breadth-first search tree was proposed to select a path on the image of an environment [19]. Jan *et al.* proposed an algorithm based on Delaunay triangulation to obtain the near-shortest path in the Euclidean plane [20]. Zhang *et al.* devised recursive receding horizon planning to find optimal path in nonuniform environments [21]. In addition, some evolutionary algorithms have been employed for path generation as well. Tsai *et al.* applied a parallel elite genetic algorithm for global path planning [22]. Analogously, Cheng *et al.* devised a genetic-algorithm-inspired unmanned underwater vehicle path planner [23].

In this paper, path generation for autonomous approach to a moving ship is conceived to produce a smooth trajectory within constraints of curvature radius and pitch angle. Many path generation issues about trajectory smoothness have been discussed so far. B-splines and Bezier curves have usually been exploited for trajectory generation considering the continuity of trajectory [24], [25]. The curve consisting of two constant curvature segments was used for path shaping technique [26]. In addition, η^4 spline was also utilized for path generation [27]. However, in 3-D space, search methods are usually applied to meet vehicle's geometric constraints. Belkhouche and Bendjilali designed a trajectory, on which the pitch angle and the heading angle are linear functions of the visibility line angles on the vertical and

Manuscript received July 27, 2014; revised December 8, 2014 and January 5, 2015; accepted January 28, 2015. Date of publication February 20, 2015; date of current version August 7, 2015. This work was supported in part by the National Natural Science Foundation of China under Grant 51175496, Grant 61333016, and Grant 61273337; by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China under Grant 61421004; and by the Beijing Natural Science Foundation under Grant 3141002.

Y. Wang is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with University of Chinese Academy of Sciences, Beijing 100049, China.

S. Wang and M. Tan are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: shuo.wang@ia.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2015.2405904

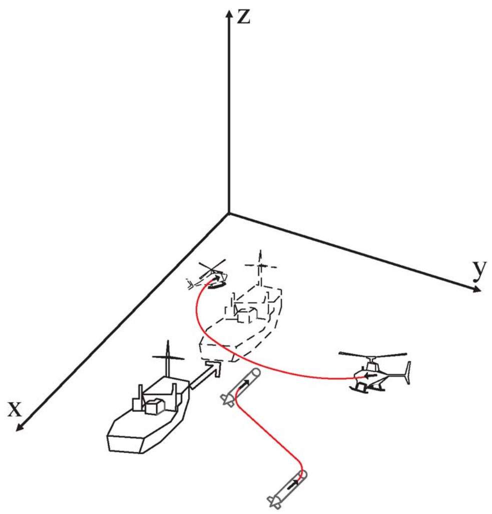


Fig. 1. Autonomous navigation of an unmanned vehicle.

horizontal planes [28]. However, the problem of autonomous navigation for approaching a moving ship was not discussed in these related work.

The main purpose of this paper is to develop an algorithm for autonomous navigation of unmanned vehicles. This algorithm is conceived to produce a trajectory for the unmanned vehicle to approach a moving mother ship before it starts automatic landing or recovery to the mother ship. More specifically, 3-D feasible trajectory, which meets geometric constraints closely related to actual ones, is designed for the autonomous approach. Moreover, the final poses the vehicle and the ship reach simultaneously are deduced, and the related parameters are discussed. Thus, autonomous navigation toward a moving ship can be realized. The notable advantage of the proposed algorithm is the real-time online planning, which is very important, particularly for the high-speed unmanned vehicle. According to the authors' knowledge, this may be the first time to apply path generation to settle this problem.

In the remainder of this paper, problem formulation is described in Section II. The approaching algorithm, including 3-D path generation for unmanned vehicles, the analysis of the final poses the ship and the vehicle reach simultaneously, and the summary of the algorithm, is elaborated in Section III. Numerical simulations are further provided in Section IV. Finally, the conclusion is presented in Section V.

II. Problem Formulation

It is assumed that there is no obstacle in the environment, and no disturbance is considered. In order to achieve the autonomous navigation of an unmanned vehicle to a moving mother ship, as described in Fig. 1, first, the feasible trajectory of the vehicle should be produced from the pose where command that a vehicle returns to the mother ship has been received to a final pose, which can provide the foundation for the vehicle's automatic landing or recovery to the mother ship. Hence, the final goal is determined: the attitudes of the vehicle and the mother ship are the same, and the vehicle is just above the mother ship with a relative height distance; the vehicle and the mother ship are sailing at the same speed.

$P_s(x_s, y_s, z_s)$ and $P_g(x_g, y_g, z_g)$ represent the vehicle initial and final positions, respectively. θ_s, γ_s are the vehicle initial heading and pitch angles, respectively; and θ_g, γ_g are the vehicle final heading and pitch angles, respectively. Suppose that the ship sails on the horizontal plane under the guidance of the planned trajectory. $P_{\text{ships}}(x_{\text{ships}}, y_{\text{ships}}, 0)$ denotes the ship initial position, and θ_{ships} denotes the heading angle of the ship. The ship speed denoted by v_{ship} is constant. The final realization of autonomous approach is expressed by

$$\begin{cases} T_v = T_s \\ x_g = x_{\text{shipg}} \\ y_g = y_{\text{shipg}} \\ z_g = h \\ \theta_g = \theta_{\text{shipg}} \\ \gamma_g = 0 \\ v_g = v_{\text{ship}} \end{cases} \tag{1}$$

where T_v and T_s are the vehicle and mother ship arrival times, respectively; h is the final relative height distance between the ship and the vehicle; and θ_{shipg} is the ship final heading angle. $P_{\text{shipg}}(x_{\text{shipg}}, y_{\text{shipg}}, 0)$ is the final ship position. v_g is the final speed of the vehicle. As the approaching trajectory needs to be produced online by the unmanned vehicle, the approaching method should satisfy real-time performance.

In a Cartesian coordinate frame (x, y, z), a smooth trajectory of an unmanned vehicle is provided according to the following equations:

$$\begin{cases} \dot{x} = v_v \cos\theta \cos\gamma \\ \dot{y} = v_v \sin\theta \cos\gamma \\ \dot{z} = v_v \sin\gamma \end{cases} \tag{2}$$

where v_v is the vehicle speed, θ is the heading angle, and γ is the pitch angle. To ensure 3-D trajectory feasible, the vehicle actual constraints should be addressed. However, in practical engineering applications, trajectory geometric constraints instead of the vehicle actual ones are usually taken into account, because trajectory geometric constraints are related to the vehicle actual ones [10]. In this paper, we design 3-D trajectory considering geometric constraints, including the curvature radius and pitch angle constraints, as follows.

1) The curvature radius constraint: $|R| \geq R_{\min}$.
2) The pitch angle constraint: $\gamma_{\min} \leq \gamma \leq \gamma_{\max}$.

Remark 1: The cooperation between the unmanned vehicle and the mother ship is essential. To accomplish a specific task, the mother ship sails to a designated pose under the guidance of the planned path. This planned path is composed of straight lines or circles. Thus, we just realize autonomous approach to a moving mother ship along a straight line or a circle. Moreover, another premise is that the positions and velocities of the unmanned vehicle and the mother ship can be measured and sent to the planner in real time.

III. Approaching Algorithm

First, 3-D path generation is designed for the autonomous approach. It can produce a feasible trajectory with geometric

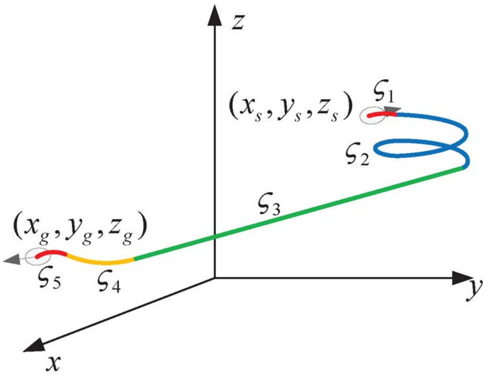


Fig. 2. Feasible trajectory for an unmanned vehicle. Subpath ς_1 is the result of pitch angle smoothing. Subpaths ς_2–ς_4 form a 3-D trajectory based on the RDDH method. Subpath ς_5 is the curve for pitch angle and speed adjustment.

constraints. The final poses the unmanned vehicle and the ship reach simultaneously are analyzed by establishing equations. This analysis is divided into two cases, where the ship path is a straight line or a circle. Hence, approaching the moving ship for an unmanned vehicle can be realized. We first introduce 3-D path generation for the navigation of the unmanned vehicle.

A. Three-Dimensional Path Generation for Unmanned Vehicle

To produce a smooth trajectory with the curvature radius and pitch angle constraints, the vehicle path is composed of five subpaths ς_1–ς_5 as follows (see Fig. 2): Subpath ς_1 is an arc with a radius of $R_{\min}$, in which the heading angle is fixed and the pitch angle changes linearly from γ_s to γ_t. γ_s is the initial pitch angle of the unmanned vehicle, and γ_t is the pitch angle of subpaths ς_2–ς_4. Subpaths ς_2–ς_4 form a 3-D trajectory based on the real-time dynamic Dubins-Helix (RDDH) method [29], which is to let the unmanned vehicle fall on the ship path while keeping a small distance S_L with the ship along the ship path on the x–y plane and a height distance h with the ship. Moreover, the vehicle's heading is consistent with the tangent angle of the ship path. The vehicle speed (denoted by v) in subpaths ς_1–ς_4 is constant, satisfying $v > v_{\text{ship}}$. S_L is reserved for pitch angle and speed adjustment; thus, subpath ς_5 is designed, in which the pitch angle and the speed are adjusted to 0 and v_{ship}, respectively; and the projection of subpath ς_5 on the x–y plane is the part of the ship path. Finally, the final poses of the unmanned vehicle and the ship can satisfy (1) under the guidance of this integrated trajectory. Next, the 3-D trajectory based on the RDDH method and the curve for pitch angle and speed adjustment are detailed, respectively.

1) Three-Dimensional Trajectory Based on RDDH Method: Subpaths ς_2–ς_4 constitute a 3-D trajectory based on the RDDH method, where the pitch angle is fixed (denoted by γ_t). The initial and final positions of ς_2–ς_4 are denoted by $P_0(x_0, y_0, z_0)$ and $P_e(x_e, y_e, z_e)$, respectively, where P_0 is the final position of subpath ς_1. θ_s and θ_e are the initial and final heading angles of subpaths ς_2–ς_4, respectively. θ_s is also the heading angle of subpath ς_1. The projection of this whole trajectory on the x–y plane is an external tangential CSC trajectory.

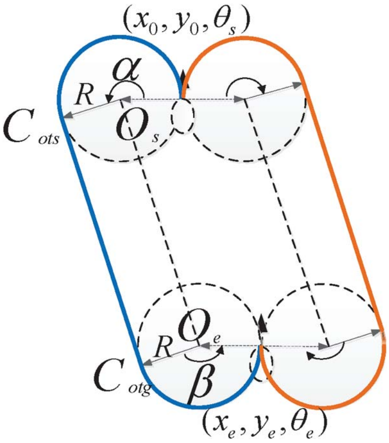


Fig. 3. 2-D external tangential CSC curves. The blue trajectory is the counterclockwise external tangential CSC curve. The yellow trajectory is the clockwise external tangential CSC curve.

Two-Dimensional External Tangential CSC Curve: There are two external tangential CSC paths, as shown in Fig. 3, and the specific description can be found in our previous work [29]. They are named the counterclockwise external tangential CSC curve (blue curve) and the clockwise external tangential CSC curve (yellow curve), respectively. Suppose that t is the parameter of selecting 2-D external tangential CSC curves. When $t = 1$, the counterclockwise external tangential CSC curve is selected for 3-D path generation, and when $t = -1$, the clockwise external tangential CSC one is selected. The path length L_{CSC} is calculated as

$$L_{\text{CSC}} = \theta_r \cdot R + \|\overrightarrow{O_sO_e}\| \tag{3}$$

where θ_r is the rotation heading angle along the whole 2-D external tangential CSC curve, which is the sum of α and β in Fig. 3, and $\|\overrightarrow{O_sO_e}\|$ is given by (4), shown at the bottom of the page.

The 2-D external tangential CSC curves have a special characteristic: the rotation heading angle θ_r is related to θ_{sg}, which is the angle between the unit vector $(\cos\theta_s, \sin\theta_s)$ and the unit vector $(\cos\theta_e, \sin\theta_e)$. To introduce this relationship, a definition is given first.

Definition 1: Suppose that $\overrightarrow{P_1}(x_1, y_1)$ and $\overrightarrow{P_2}(x_2, y_2)$ are two unit vectors, the relative relationship between the two vectors $\overrightarrow{P_1}$ and $\overrightarrow{P_2}$ can be defined into three cases. As shown in Fig. 4, when $\overrightarrow{P_2}$ is the red and dotted vector, $y_1x_2 - y_2x_1 < 0$, we define that $\overrightarrow{P_2}$ is in the counterclockwise direction of the vector $\overrightarrow{P_1}$. When $\overrightarrow{P_2}$ is the blue and solid vector, $y_1x_2 - y_2x_1 > 0$, we define that $\overrightarrow{P_2}$ is in the clockwise direction of $\overrightarrow{P_1}$. When $y_1x_2 - y_2x_1 = 0$, $\overrightarrow{P_2}$ and $\overrightarrow{P_1}$ are the same or opposite vectors.

$$\|\overrightarrow{O_sO_e}\| = \sqrt{(x_e - tR\sin\theta_e - x_0 + tR\sin\theta_s)^2 + (y_e + tR\cos\theta_e - y_0 - tR\cos\theta_s)^2} \tag{4}$$

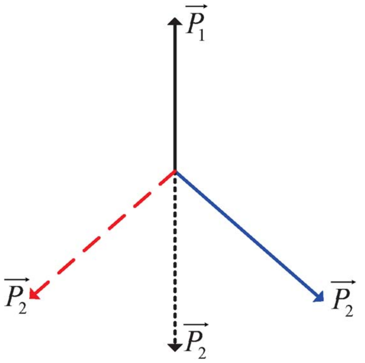


Fig. 4. Relative relationship between two unit vectors.

The relationship between θ_r and θ_{sg} is discussed based on *Definition 1*. It is divided into four cases, as depicted in Fig. 5. The rotation heading angle is a function about θ_{sg}, t, s, which is expressed as

$$\theta_r = f(\theta_{sg}, t, s) \tag{5}$$

where s is the internal parameter determined by the relative relationship between the unit vector (q_{ox}, q_{oy}) and the unit vector $(\cos\theta_e, \sin\theta_e)$ as

$$s = \begin{cases} 1, & \sin\theta_e q_{ox} - \cos\theta_e q_{oy} \geq 0 \\ -1, & \sin\theta_e q_{ox} - \cos\theta_e q_{oy} < 0 \end{cases} \tag{6}$$

where (q_{ox}, q_{oy}) is the unit vector of $\overrightarrow{O_s O_e}$, as illustrated in Fig. 5.

Taking the case when $t=1$, $s=1$ for an example, the specific expression of the relationship between θ_r and θ_{sg} is given. As illustrated in Fig. 5(a), when the vector $(\cos\theta_s, \sin\theta_s)$ is in the counterclockwise direction of $(\cos\theta_e, \sin\theta_e)$ (see the purple circle), $\theta_r = 2\pi - \theta_{sg}$. When the vector $(\cos\theta_s, \sin\theta_s)$ is in the clockwise direction of $(\cos\theta_e, \sin\theta_e)$ and in the clockwise direction of (q_{ox}, q_{oy}) (see the red circle), $\theta_r = \theta_{sg}$. When the vector $(\cos\theta_s, \sin\theta_s)$ is in the clockwise direction of $(\cos\theta_e, \sin\theta_e)$ and in the counterclockwise direction of (q_{ox}, q_{oy}) (see the blue circle), $\theta_r = 2\pi + \theta_{sg}$. Hence, according to the four cases in Fig. 5, the relationship between θ_r and θ_{sg} can be given by

$$\theta_r = \begin{cases} 2\pi - s \cdot t \cdot \theta_{sg}, & c_0 \leq 0 \\ 2\pi - s \cdot t \cdot (2\pi - \theta_{sg}), & c_0 > 0 \text{ and } c_1 \geq 0 \\ 2\pi + s \cdot t \cdot \theta_{sg}, & c_0 > 0 \text{ and } c_1 < 0 \end{cases} \tag{7}$$

where $c_0 = s(\sin\theta_e \cos\theta_s - \sin\theta_s \cos\theta_e)$, and $c_1 = s(q_{oy}\cos\theta_s - \sin\theta_s q_{ox})$.

In light of (7), we can summarize: Once the parameters t and s are determined, the relationship is divided into three cases, and in each case, the rotation heading angle θ_r is a fixed value related to θ_{sg}. This special property is important to derive the equations computing the final pose in Section III-B.

Three-Dimensional Extension: The 3-D trajectory based on the RDDH method is a 3-D extension whose pitch angle γ_t is fixed. It is composed of three subpaths ς_2–ς_4. As shown in Fig. 2, the projection of the subpaths ς_2–ς_4 on the x–y plane is an external tangential CSC trajectory, and the curvature radius on the x–y plane is $R_{\min}$. The subpaths ς_2 and ς_4 are helix curves. Subpath ς_3 is the straight line. The projection of the helix curve on the x–y plane is a circle of radius $R_{\min}$. The pitch angle of the helix curve is γ_t. In subpath ς_2, the rotation heading angle θ_2 on the x–y plane is $\alpha + 2\pi N$, where N is the number of rotational laps. In subpath ς_4, the rotation heading angle θ_4 is β. In order to satisfy the pitch angle constraint, the trajectory is divided into two cases according to the relationship between $L_{\mathrm{CSC}}\tan\gamma_m$, and $|z_e - z_0|$, where γ_m is defined as

$$\gamma_m = \begin{cases} \gamma_{\max}, & z_e - z_0 \geq 0 \\ |\gamma_{\min}|, & z_e - z_0 < 0. \end{cases} \tag{8}$$

N and γ_t can be respectively obtained as

$$N = \begin{cases} 0, & L_{\mathrm{CSC}} \cdot \tan\gamma_m \geq |z_e - z_0| \\ \left\lceil \frac{|z_e - z_0| - L_{\mathrm{CSC}}\tan\gamma_m}{2\pi R_{\min}\tan\gamma_m} \right\rceil, & L_{\mathrm{CSC}} \cdot \tan\gamma_m < |z_e - z_0| \end{cases} \tag{9}$$

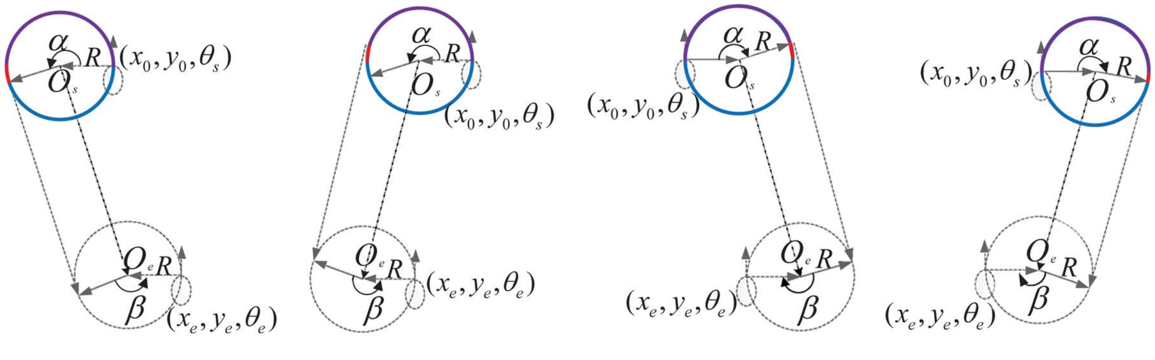
$$\gamma_t = \arctan\frac{z_e - z_0}{L_{\mathrm{CSC}} + 2\pi R_{\min} N}. \tag{10}$$

2) Curve for Pitch Angle and Speed Adjustment: Subpath ς_5 is the curve for pitch angle and speed adjustment. The projection of subpath ς_5 on the x–y plane is a part of the ship path, which is a straight line or a circle. As depicted in Fig. 6, the red curve is the $S - z$ diagram of subpath ς_5, where S is the abscissa along the projection of subpath ς_5 on the x–y plane (a straight line or a circle). $P_e(x_e, y_e, z_e)$ is the initial position of ς_5, which is also the final position of 3-D trajectory based on the RDDH method. θ_e is the initial heading angle of subpath ς_5. P_{shipe} represents the ship position when the unmanned vehicle reaches P_e. θ_{shipe} represents the ship heading angle when the ship reaches P_{shipe}. P_m represents the vehicle final position of pitch angle adjustment, which is also the initial position of speed adjustment. P_{shipm} represents the ship position when the unmanned vehicle reaches P_m. The red curve between P_e and P_m is the process of pitch angle adjustment, where the vehicle speed is v, whereas the other red one between P_m and P_g is the process of speed adjustment from v to v_{ship} at a fixed acceleration a_v. S_{vp} is the S value between P_e and P_m, S_{sp} is the sailing distance of the ship during pitch angle adjustment, S_{vv} is the S value between P_m and P_g, and S_{sv} is the sailing distance of the ship during speed adjustment. Fig. 6 shows the description of subpath ς_5. χ_1 is a circle of radius $R_{\min}$ and intersects P_e, and the tangent direction of χ_1 at P_e is determined by γ_t. χ_2 is a circle of radius $R_{\min}$ and intersects both χ_1 and $z = h$ at exactly one point each. The tangent point between the circle χ_2 and the line $z = h$ is P_m. Hence, we can get

$$S_L = S_{vp} + S_{vv} - S_{sp} - S_{sv} \tag{11}$$

where

$$\begin{aligned} S_{vp} &= \sqrt{4R_{\min}^2 - (R_{\min}\cos\gamma_t + R_{\min})^2} + R_{\min}\sin\gamma_t \\ S_{sp} &= v_{\mathrm{ship}} \frac{\left(2\arccos\left(\frac{\cos\gamma_t + 1}{2}\right) + \gamma_t\right) R_{\min}}{v} \\ S_{vv} &= \frac{v^2 - v_{\mathrm{ship}}^2}{2a_v} \\ S_{sv} &= v_{\mathrm{ship}} \cdot \frac{v - v_{\mathrm{ship}}}{a_v}. \end{aligned} \tag{12}$$



(a) (b) (c) (d)

Fig. 5. Classification of the computation of θ_r. (a) $s = 1, t = 1$. (b) $s = -1, t = 1$. (c) $s = 1, t = -1$. (d) $s = -1, t = -1$.

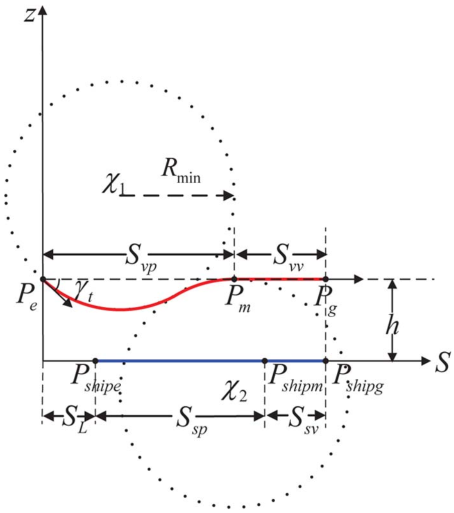


Fig. 6. $S - z$ diagram of subpath ς_5.

As a consequence, the integrated trajectory, which consists of pitch angle smoothing (subpath ς_1), the trajectory based on the RDDH method (subpaths ς_2–ς_4), and the curve for pitch angle and speed adjustment(subpath ς_5), is designed for autonomous navigation of the unmanned vehicle. This trajectory can guide the unmanned vehicle to approach the moving mother ship and provide a good condition for automatic landing or recovery to the mother ship [satisfying (1)].

B. Analysis of Final Poses Unmanned Vehicle and Ship Reach Simultaneously

In order to let the vehicle and the ship arrive simultaneously at their respective poses, the related equations are established based on 3-D path generation. The equations are the constrained quadratic or linear ones when the ship sails along a straight line, and the final pose can be obtained in an analytic form. When the ship sails along a circle, the nonlinear equations are obtained. They can be solved via numerical methods.

1) Approaching Ship Moving Along a Straight Line: Before the approach to a moving ship starts, the vehicle pitch angle should be adjusted to an initial pitch angle. Thus, the vehicle sailing time on subpath ς_1, which is the result of the pitch angle smoothing process, can be negligible. Thus, we can get $P_0(x_0, y_0, z_0) = P_s(x_s, y_s, z_s)$. In the case of $\cos\theta_{\text{ships}} \neq 0$, let $T_v = T_s$, we can get

$$\frac{\theta_r R_{\min} + \|\overrightarrow{O_s O_e}\| + 2\pi R_{\min} N}{v \cos \gamma_t} = \frac{x_e + S_L \cos\theta_{\text{ships}} - x_{\text{ships}}}{v_{\text{ship}} \cos\theta_{\text{ships}}}. \tag{13}$$

By substituting (4) into (13), one linear equation can be obtained as

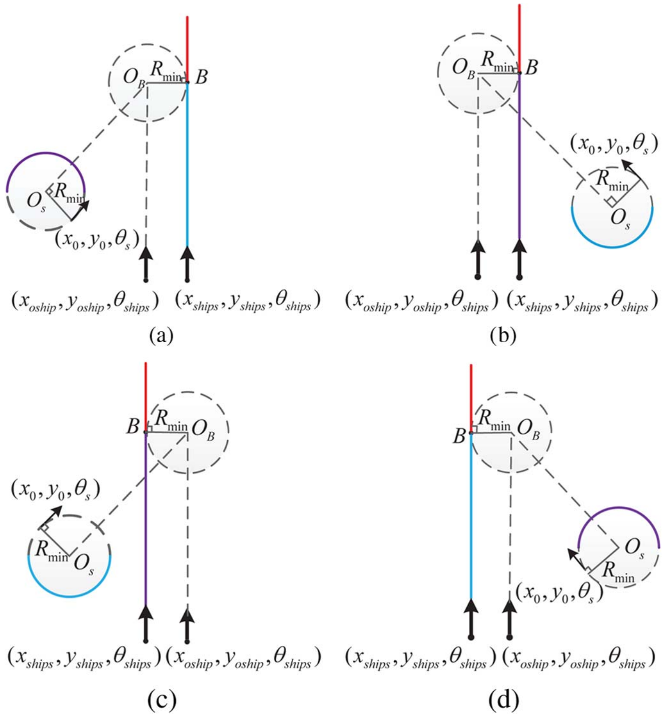
$$\begin{cases} (1 + k^2 - l^2)x_e^2 + (2A + 2kB + 2lC)x_e \\ +A^2 + B^2 - C^2 = 0 \\ x_e \geq \frac{C}{l} \end{cases} \tag{14}$$

where

$$\begin{aligned} k &= \tan\theta_{\text{ships}} \\ l &= \frac{v \cos\gamma_t}{(v_{\text{ship}} \cdot \cos\theta_{\text{ships}})} \\ A &= -x_s + tR_{\min}(\sin\theta_s - \sin\theta_{\text{ships}}) \\ B &= y_{\text{ships}} - y_s - kx_{\text{ships}} + tR_{\min}(\cos\theta_{\text{ships}} - \cos\theta_s) \\ C &= lx_{\text{ships}} + \theta_r R_{\min} + 2\pi R_{\min} N - lS_L \cos\theta_{\text{ships}}. \end{aligned} \tag{15}$$

In the other case of $\cos\theta_{\text{ships}} = 0$, the derivation of the equation is similar to (14) and thus omitted. Due to the uncertainty of the final position, three unknown variables need to be determined in the equations: the pitch angle γ_t, the number of rotational laps N, and the rotation heading angle θ_r. γ_t can be obtained by a simple search method. According to our experience, γ_t can be computed after less than ten iterations. Due to (9), N is a decreasing function about the projection length L_{CSC} of the trajectory on the x–y plane. Hence, the maximum value $N_{\max}$ is obtainable; we can get the solution of (14) using a limited and small number of cycles meeting $N \leq N_{\max}$. Here, we only introduce the calculations of θ_r using the special characteristic of 2-D external tangential CSC curves.

As depicted in Fig. 7, the computation of θ_r is discussed in four cases according to the selection of the parameters t and s. The ship sails along a straight line; thus, the final attitudes of


O_B
$R_{\min}$
B
O_s
$R_{\min}$
(x_0, y_0, θ_s)
$(x_{oship}, y_{oship}, \theta_{ships})$
$(x_{ships}, y_{ships}, \theta_{ships})$

(a) (b) (c) (d)

Fig. 7. Classification of the computation of θ_r when the ship sails along a straight line. (a) $s = 1, t = 1$. (b) $s = -1, t = 1$. (c) $s = 1, t = -1$. (d) $s = -1, t = -1$.

the ship and the vehicle are fixed. Therefore, we can get s by (16), which is equivalent to

$$s = \begin{cases} 1, & \sin\theta_{\text{ships}} q_{ox} - \cos\theta_{\text{ships}} q_{oy} \geq 0 \\ -1, & \sin\theta_{\text{ships}} q_{ox} - \cos\theta_{\text{ships}} q_{oy} < 0 \end{cases}$$

$$s = \begin{cases} 1 & \cos\theta_{\text{ships}} \\ & \times [y_{os} - y_{\text{oship}} - \tan\theta_{\text{ships}}(x_{os} - x_{\text{oship}})] \geq 0 \\ -1 & \cos\theta_{\text{ships}} \\ & \times [y_{os} - y_{\text{oship}} - \tan\theta_{\text{ships}}(x_{os} - x_{\text{oship}})] < 0 \end{cases} \tag{16}$$

where $\binom{x_{\text{oship}}}{y_{\text{oship}}} = \binom{x_{\text{ships}} - t \cdot R_{\min} \cdot \sin\theta_{\text{ships}}}{y_{\text{ships}} + t \cdot R_{\min} \cdot \cos\theta_{\text{ships}}}$, and $\binom{x_{os}}{y_{os}} = \binom{x_s - t \cdot R_{\min} \cdot \sin\theta_s}{y_s + t \cdot R_{\min} \cdot \cos\theta_s}$; from (16), we can see that once 2-D external tangential CSC curve is selected (namely, t is determined), the parameter s is determined.

We utilize (7) to obtain the rotation heading angle θ_r. As shown in Fig. 7, in the case of $s \cdot (\sin\theta_{\text{ships}} \cos\theta_s - \cos\theta_{\text{ships}} \sin\theta_s) \leq 0$, $\theta_r = 2\pi - s \cdot t \cdot \theta_{sg}$, else another two cases need to be divided. $B(x_B, y_B)$ in Fig. 7 is the boundary of the two cases, and the vectors $\overrightarrow{O_s O_B}$ and $(\cos\theta_s, \sin\theta_s)$ are in the same or opposite directions. Thus, in the other case of $s \cdot (\sin\theta_{\text{ships}} \cos\theta_s - \cos\theta_{\text{ships}} \sin\theta_s) > 0$, if $\cos\theta_{\text{ships}}(x_{\text{B}} - x_{\text{ships}}) \leq 0$, $\theta_r = 2\pi - s \cdot t \cdot (2\pi - \theta_{sg})$, else we can obtain

$$\theta_r = \begin{cases} 2\pi - s \cdot t \cdot (2\pi - \theta_{sg}), & c_2 \geq 0 \\ 2\pi + s \cdot t \cdot \theta_{sg}, & c_2 < 0 \text{ and } c_3 \geq 0 \end{cases} \tag{17}$$

where $c_2 = \cos\theta_{\text{ships}}(x_e - x_B)$, and $c_3 = \cos\theta_{\text{ships}}(x_e - x_{\text{ships}})$.

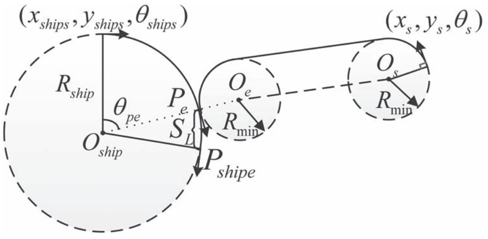


Fig. 8. Analysis of the final goal when the ship sails along a clockwise circle. The circle that the ship sails along intersects the final arc of the 2-D external tangential CSC curve at exactly one point. P_e is the tangent point between the above two circles.

Therefore, the final poses the ship and the vehicle reach simultaneously can be obtained once (14) is solved.

2) Approaching Ship Moving Along a Circle: Here, we just take the case where the ship sails along a clockwise circle for an example to derive the equations. As shown in Fig. 8, the circle centers $O_s(x_{os}, y_{os})$ and $O_e(x_{oe}, y_{oe})$ can be obtained

$$\begin{bmatrix} x_{os} \\ y_{os} \end{bmatrix} = R_{\min} \cdot \begin{bmatrix} \cos\left(\frac{\pi}{2} \cdot t\right) & -\sin\left(\frac{\pi}{2} \cdot t\right) \\ \sin\left(\frac{\pi}{2} \cdot t\right) & \cos\left(\frac{\pi}{2} \cdot t\right) \end{bmatrix} \begin{bmatrix} \cos\theta_s \\ \sin\theta_s \end{bmatrix} + \begin{bmatrix} x_s \\ y_s \end{bmatrix} \tag{18}$$

$$\begin{bmatrix} x_{oe} \\ y_{oe} \end{bmatrix} = (R_{\text{ship}} + tR_{\min}) \cdot \begin{bmatrix} \cos(-\theta_{pe}) & -\sin(-\theta_{pe}) \\ \sin(-\theta_{pe}) & \cos(-\theta_{pe}) \end{bmatrix} \times \begin{bmatrix} -\sin\theta_{\text{ships}} \\ \cos\theta_{\text{ships}} \end{bmatrix} + \begin{bmatrix} x_{\text{oship}} \\ y_{\text{oship}} \end{bmatrix} \tag{19}$$

where $O_{\text{ship}}(x_{\text{oship}}, y_{\text{oship}})$ and R_{ship} are the center and the radius of the circle that the ship sails along, respectively; and θ_{pe} is marked in Fig. 8.

Let $T_v = T_s$, we can get

$$\frac{\theta_r R_{\min} + \|\overrightarrow{O_s O_e}\| + 2\pi R_{\min} N}{v \cos\gamma_t} = \frac{\theta_{pe} R_{\text{ship}} + S_L}{v_{\text{ship}}}. \tag{20}$$

By substituting (18) and (19) into (20), we can obtain

$$\sqrt{D + E\sin\theta_{pe} + F\cos\theta_{pe}} = \frac{v\cos\gamma_t \theta_{pe} R_{\text{ship}}}{v_{\text{ship}}} - \theta_r R_{\min} - 2\pi R_{\min} N + \frac{v S_L \cos\gamma_t}{v_{\text{ship}}} \tag{21}$$

where

$$\begin{aligned} D =& R_s^2 + (x_{\text{oship}} - x_{os})^2 + (y_{\text{oship}} - y_{os})^2 \\ E =& 2R_s \cdot [(x_{\text{oship}} - x_{os}) \cdot \cos\theta_{\text{ships}} \\ &+ (y_{\text{oship}} - y_{os}) \cdot \sin\theta_{\text{ships}}] \\ F =& 2R_s \cdot [-(x_{\text{oship}} - x_{os}) \cdot \sin\theta_{\text{ships}} \\ &+ (y_{\text{oship}} - y_{os}) \cdot \cos\theta_{\text{ships}}] \\ R_s =& R_{\text{ship}} + tR_{\min}. \end{aligned} \tag{22}$$

The equations can be solved by numerical methods. In the equations, the pitch angle γ_t can be obtained by a search method, and the maximum of N can be also determined. Here, the computation of θ_r is described.

We also employ the special characteristic of 2-D external tangential CSC curves to obtain θ_r, which is discussed in two cases.

When $\sin\theta_{\text{ships}}\cos\theta_s - \cos\theta_{\text{ships}}\sin\theta_s < 0$, θ_r can be obtained as

$$\theta_r = \begin{cases} 2\pi - t\cdot\theta_{sg} \\ \quad \text{or } 2\pi + t\cdot(2\pi-\theta_{sg}), & \theta_{pe}\in[0,\pi-\theta_L] \\ 2\pi - t\cdot(2\pi-\theta_{sg}) \\ \quad \text{or } 2\pi + t\cdot\theta_{sg}, & \theta_{pe}\in(\pi-\theta_L, 2\pi-\theta_L] \\ 2\pi - t\cdot\theta_{sg} \\ \quad \text{or } 2\pi + t\cdot(2\pi-\theta_{sg}), & \theta_{pe}\in(2\pi-\theta_L, 2\pi] \end{cases} \tag{23}$$

where θ_{sg} is computed as

$$\theta_{sg} = \begin{cases} \theta_L + \theta_{pe}, & \theta_{pe}\in[0,\pi-\theta_L] \\ 2\pi - \theta_{pe} - \theta_L, & \theta_{pe}\in(\pi-\theta_L, 2\pi-\theta_L] \\ \theta_{pe} + \theta_L - 2\pi, & \theta_{pe}\in(2\pi-\theta_L, 2\pi] \end{cases} \tag{24}$$

where θ_L is the angle between the vectors $(\cos\theta_s, \sin\theta_s)$ and $(\cos\theta_{\text{ships}}, \sin\theta_{\text{ships}})$.

When $\sin\theta_{\text{ships}}\cos\theta_s - \cos\theta_{\text{ships}}\sin\theta_s \geq 0$, θ_r can be obtained as

$$\theta_r = \begin{cases} 2\pi + t\cdot\theta_{sg} \\ \quad \text{or } 2\pi - t\cdot(2\pi-\theta_{sg}), & \theta_{pe}\in[0,\theta_L] \\ 2\pi - t\cdot\theta_{sg} \\ \quad \text{or } 2\pi + t\cdot(2\pi-\theta_{sg}), & \theta_{pe}\in(\theta_L, \pi+\theta_L] \\ 2\pi + t\cdot\theta_{sg} \\ \quad \text{or } 2\pi - t\cdot(2\pi-\theta_{sg}), & \theta_{pe}\in(\pi+\theta_L, 2\pi] \end{cases} \tag{25}$$

where θ_{sg} is computed as

$$\theta_{sg} = \begin{cases} \theta_L - \theta_{pe}, & \theta_{pe}\in[0,\theta_L] \\ \theta_{pe} - \theta_L, & \theta_{pe}\in(\theta_L, \pi+\theta_L] \\ 2\pi - \theta_{pe} + \theta_L, & \theta_{pe}\in(\pi+\theta_L, 2\pi]. \end{cases} \tag{26}$$

The calculation of the equation for the situation where the ship sails along a counterclockwise circle is similar to the above process and thus omitted. It is noted that θ_r is different when θ_{pe} is in different areas. From (23) and (25), θ_r is calculated in three cases; thus, the final pose has to be obtained according to the three cases as well.

C. Summary

The autonomous navigation of the vehicle to a moving ship is divided into two cases according to the ship path. It is summarized in Algorithm 1. γ_t is obtained by a simple search method. In our experience, γ_t can be determined for less than ten iterations. There are two kinds of trajectories in Fig. 3, which can be selected for the projection of subpaths ς_2–ς_4 on the x–y plane. Hence, we choose the CSC trajectory that can let the vehicle and the ship spend shorter time to arrive simultaneously at their respective poses. In the case where the equations have no solution, the curvature radius is expanded to find the solutions.

Algorithm 1 Navigation of the Vehicle to a Moving Ship

```
Determine the ship path (a straight line or a circle);
Determine the maximum rotational laps number $N_{\max}$ according to the ship path;
$\gamma_t = \gamma_s$;
while (1) do
    for $N = 1 \dots N_{max}$ do
        if the ship path is a straight line then
            Compute $\theta_r$ and $S_L$;
            Compute $x_e$ by (14);
            Get the position $P_e$ and the final position $P_g$;
        else
            Compute $\theta_r$ and $S_L$;
            Compute $\theta_{pe}$ by (21) via the numerical method;
            Get the position $P_e$ and the final position $P_g$;
        end if
    end for
    Select the final position $P_g$ arrived at in a shorter time;
    Get the pitch angle $\gamma$ of the planned trajectory by (10);
    if $|\gamma - \gamma_t| < \delta$ then
        break;
    else
        $\gamma_t = \gamma$;
    end if
end while
Get the 3-D trajectory from the initial pose to the final pose.
```

TABLE I

INITIAL CONFIGURATIONS OF UNMANNED VEHICLE

	$P_s(x_s, y_s, z_s)$	θ_s	γ_s
1	$(1000, 1000, 300)$	45°	−7.5°
2	$(-1000, 1000, 300)$	225°	−7.5°
3	$(-1000, -1000, 300)$	180°	−7.5°
4	$(1000, -1000, 300)$	300°	−7.5°

IV. SIMULATION RESULTS AND DISCUSSIONS

A. Experimental Setup and Test of Real-Time Performance

The approaching algorithm is evaluated by numerical simulations. The initialization of the specific parameters for simulations is given: $v = 15$ m/s, $v_{\text{ship}} = 10$ m/s, $h = 30$ m, $P_{\text{ships}} = (0, 0, 0)$, $\theta_{\text{ships}} = 0°$, and $a_v = 5$ m/s^2. Moreover, the geometric constraints are selected: $-15° \leq \gamma \leq 15°$, $R \geq 60$ m.

Our algorithm is implemented under different initial conditions. The vehicle initial configurations are listed in Table I. The simulations are discussed in two cases, where the ship sails along a straight line and a clockwise circle, with $R_{\text{ship}} = 800$ m. The simulation of the approaching algorithm is done in 3.4-GHz Pentium-class computer with 4-G random access memory.

By analyzing the time of the simultaneous arrival, (14) and (21) are established, where only γ_t is determined by searching, but the number of iterations is limited. Moreover, time spent on solving the two equations is microsecond class once all the

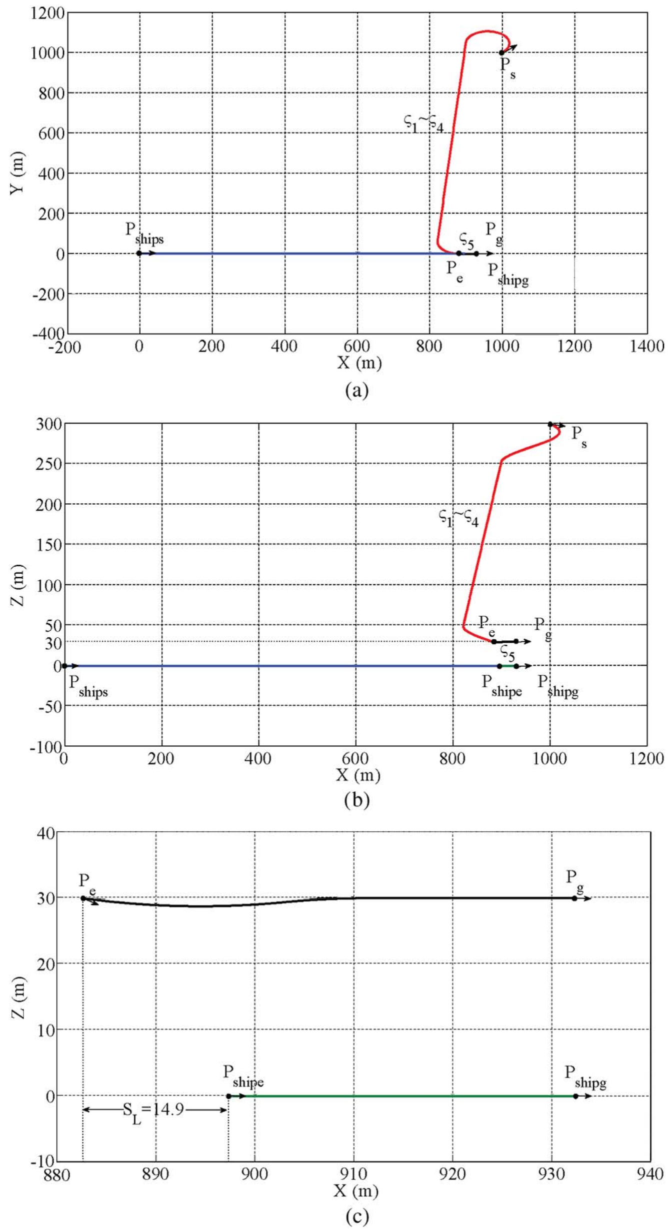


Fig. 9. Description of the approaching trajectory. (a) Vehicle and ship trajectories projected on the x–y plane. (b) Vehicle and ship trajectories projected on the x–z plane. (c) x–z diagram of subpath ς_5.

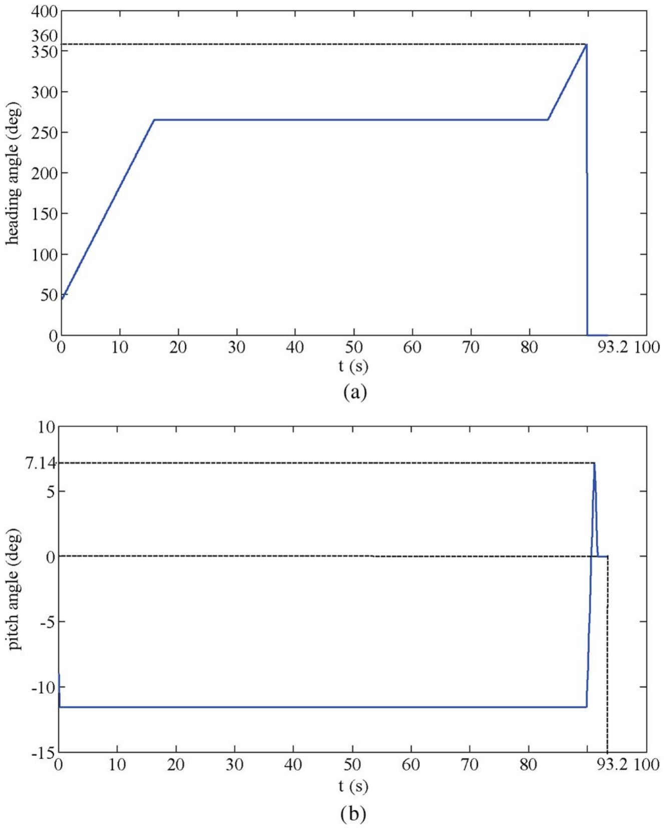


Fig. 10. Description of trajectory parameters. (a) Time history of heading angle. (b) Time history of pitch angle.

parameters are known. Thus, the proposed algorithm satisfies high real-time demand. The test is implemented to show this performance. The run average execution time is $44.9\ \mu$s in the case where the ship sails along a straight line, and the run average execution time in the other case where the ship sails along a circle is $194.1\ \mu$s. It is demonstrated that the approaching algorithm could be implemented in real time.

B. Test of Approaching Algorithm

When the ship sails along a straight line and the data in the first line in Table I are selected as the initial configurations of the vehicle, the 3-D trajectory is generated, which can guide the unmanned vehicle to a moving ship [see Fig. 11(a)]. Fig. 9(a) shows the vehicle and ship paths projected on the x–y plane, and Fig. 9(b) shows the vehicle and ship paths projected on the x–z plane, revealing that ς_1–ς_4 (see the red curve) are produced to let the unmanned vehicle fall on the ship path while keeping a small distance $S_L = 14.9$ m with the ship on the x–y plane and a height distance 30 m with the ship. Fig. 9(c) magnifies the x–z diagram of subpath ς_5 (see the black curve), which is utilized to realize pitch angle adjustment and speed adjustment. It is noted that the sailing of the trajectory, including ς_1–ς_4, needs 89.7 s to complete, whereas the sailing of ς_5 for pitch angle adjustment and speed adjustment only needs 3.5 s. Subpath ς_5 is particularly designed for the final purpose that the vehicle sails along with the ship at the same speed while keeping the same attitudes. This can provide a good condition for automatic landing or recovery to the moving ship. To verify the feasibility of the vehicle trajectory, Fig. 10 shows the time history of the vehicle pitch angle and heading angle along the whole trajectory. It is seen that the unmanned vehicle trajectory is smooth in terms of the heading angle and the pitch angle. Furthermore, the generated trajectory can meet the pitch angle constraint due to the design of the helix curve, as shown in Fig. 10(b).

To further confirm the practicality of the approaching method, simulations under the four different vehicle initial conditions in Table I are conducted. The approaching algorithm is tested in the cases where the ship sails along a straight line [see Fig. 11(a)–(d)] and a clockwise circle of $R_{\text{ship}} = 800$ [see Fig. 11(e)–(h)]. Finally, the vehicle and ship final poses can satisfy (1), whether the ship sails along a straight line or a circle. The related parameters of the paths are listed in Tables II and III.

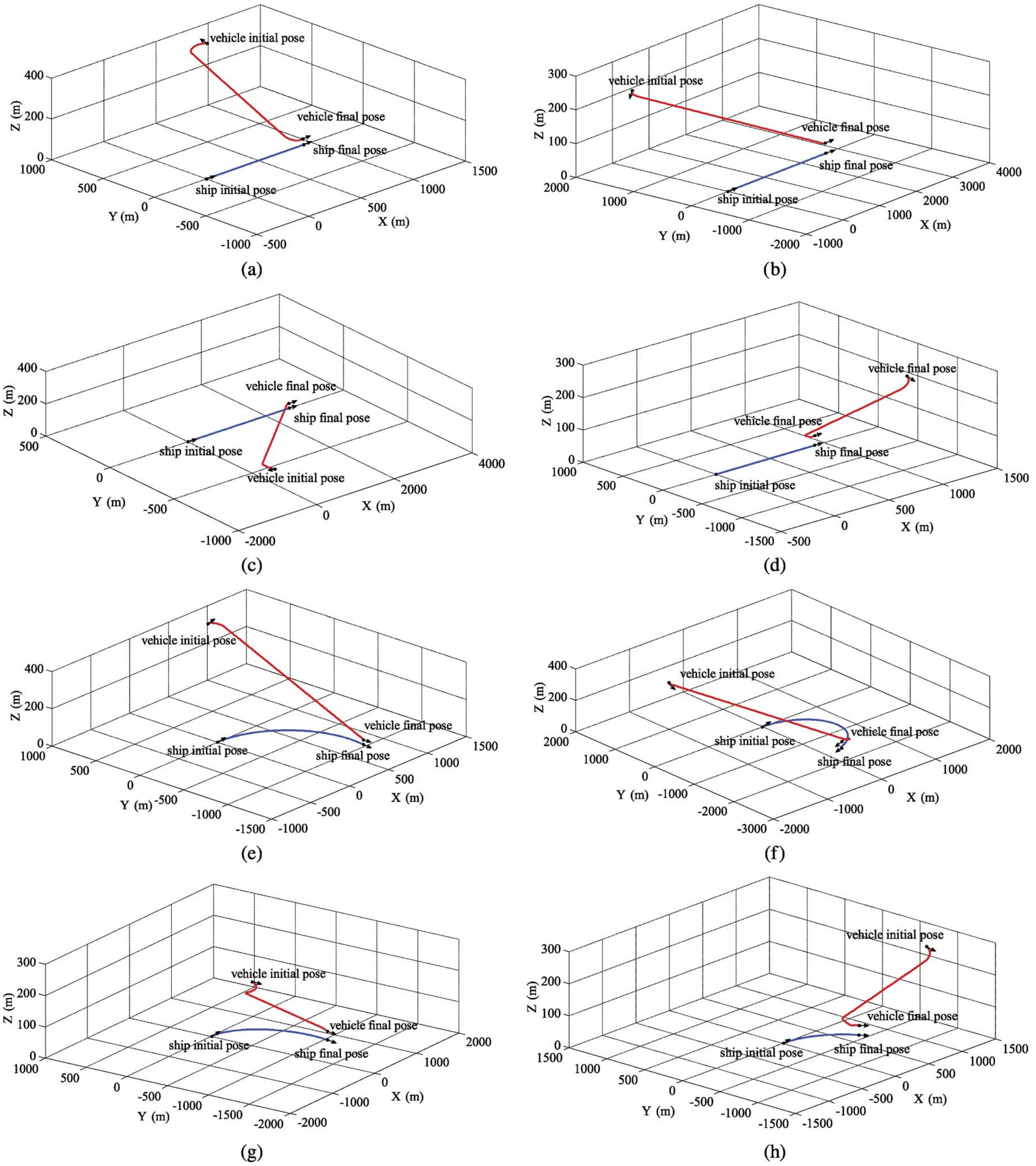


Fig. 11. Simulations under different initial conditions. (a)–(d) Test of autonomous approach to the ship along a straight line path. (e)–(h) Test of autonomous approach to the ship along a circle.

TABLE II
PARAMETERS OF POSITIONS AND ATTITUDES WHEN SHIP SAILS ALONG A STRAIGHT LINE

	P_e	θ_e	γ_t	P_{shipe}	θ_{shipe}	P_g	θ_g	γ_g	P_{shipg}	θ_{shipg}
1	$(882.6, 0, 30)$	$0°$	$-11.5°$	$(897.5, 0, 0)$	$0°$	$(932.4, 0, 30)$	$0°$	$0°$	$(932.4, 0, 0)$	$0°$
2	$(2641.0, 0, 30)$	$0°$	$-3.9°$	$(2651.7, 0, 0)$	$0°$	$(2677.9, 0, 30)$	$0°$	$0°$	$(2677.9, 0, 0)$	$0°$
3	$(2562.4, 0, 30)$	$0°$	$-4.0°$	$(2573.1, 0, 0)$	$0°$	$(2599.7, 0, 30)$	$0°$	$0°$	$(2599.7, 0, 0)$	$0°$
4	$(867.3, 0, 30)$	$0°$	$-11.7°$	$(880.0, 0, 0)$	$0°$	$(910.6, 0, 30)$	$0°$	$0°$	$(910.6, 0, 0)$	$0°$

TABLE III
PARAMETERS OF POSITIONS AND ATTITUDES WHEN THE SHIP SAILS ALONG A CIRCLE

	P_e	θ_e	γ_t	P_{shipe}	θ_{shipe}	P_g	θ_g	γ_g	P_{shipg}	θ_{shipg}
1	$(784.6, -956.4, 30)$	$258.7°$	$-7.3°$	$(783.3, -962.4, 0)$	$258.3°$	$(779.7, -979.3, 30)$	$257.1°$	$0°$	$(779.7, -979.3, 0)$	$257.1°$
2	$(362.7, -1513.0, 30)$	$207.0°$	$-4.8°$	$(347.0, -1520.8, 0)$	$205.7°$	$(313.5, -1543.3, 30)$	$202.8°$	$0°$	$(313.5, -1543.3, 0)$	$202.8°$
3	$(794.2, -896.2, 30)$	$263.1°$	$-7.6°$	$(793.0, -906.0, 0)$	$262.4°$	$(789.2, -930.8, 30)$	$260.6°$	$0°$	$(789.2, -930.8, 0)$	$260.6°$
4	$(647.4, -329.9, 30)$	$306.0°$	$-13.2°$	$(662.9, -352.1, 0)$	$304.0°$	$(694.4, -402.7, 30)$	$299.8°$	$0°$	$(694.4, -402.7, 0)$	$299.8°$

TABLE IV
COMPARISONS OF PROPOSED METHOD AND NUMERICAL METHOD [30] WHEN SHIP SAILS ALONG A STRAIGHT LINE

	Proposed method		Numerical method [30]	
	T_v	T_a	T_v	T_a
1	$93.2s$	$41.7\mu s$	$89.1s$	$2.6min$
2	$267.8s$	$47.1\mu s$	$260.3s$	$8.6min$
3	$260.0s$	$47.6\mu s$	$255.4s$	$8.2min$
4	$91.1s$	$43.3\mu s$	$86.6s$	$3.1min$

TABLE V
COMPARISONS OF PROPOSED METHOD AND NUMERICAL METHOD [30] WHEN SHIP SAILS ALONG A CIRCLE

	Proposed method		Numerical method [30]	
	T_v	T_a	T_v	T_a
1	$143.7s$	$248.2\mu s$	$136.2s$	$4.2min$
2	$219.5s$	$187.7\mu s$	$211.7s$	$5.8min$
3	$138.8s$	$153.1\mu s$	$137.2s$	$4.4min$
4	$84.1s$	$187.2\mu s$	$78.9s$	$3.6min$

Meanwhile, the comparisons of the proposed method and the numerical method named multiple shooting [30] under the vehicle initial configurations in Table I are given. Two methods are evaluated by the time of simultaneous arrival (T_v) and the execution time of the method (denoted by T_a). Note that the multiple shooting method can generate an optimal path between an initial pose and a final pose, which can meet the curvature radius and pitch angle constraints [30]. We use a simple binary method to search T_v for satisfying $T_v = T_s$. There are only several iterations to get the arrival time by the binary method, but it takes a relatively long time for the multiple shooting method to produce an optimal path from an initial pose to a fixed final pose [30]. The arrival times calculated by the proposed method and the numerical method and the execution times of the two methods are tabulated in Tables IV and V. The arrival time computed by our method is slightly larger than that computed by the numerical method. However, the final goal of the approaching method is to provide a good condition for automatic landing or recovery, and the approaching path needs to be planned online. Thus, the path that guides the unmanned vehicle to the moving ship should be generated in high real time, particularly for the high-speed unmanned vehicle. Under the same configurations (see Tables IV and V), the average execution time of the numerical method in [30] is minute class, whereas the average execution time of the proposed method is microsecond class, far less than that of the numerical method.

From the comparisons in Tables IV and V, we can summarize that the arrival time computed by the proposed method is acceptable compared with that by the numerical method. Furthermore, the proposed method can satisfy high real-time demand, which can generate an approaching path online.

Note that it is important to select suitable parameters of geometric constraints for path generation, because geometric constraints, including curvature radius and pitch angle, are closely related to the actual ones. If geometric parameters are improper, big tracking errors would occur. Moreover, once big errors occur, path replanning can be implemented online due to the real-time performance of the proposed method. The future work will popularize this method for practical applications.

V. CONCLUSION

In this paper, we have presented an approaching method based on path generation for autonomous navigation of an unmanned vehicle to a moving mother ship. It can satisfy high real-time demand. Path generation provides a smooth trajectory within constraints of curvature radius and pitch angle for the unmanned vehicle. Autonomous navigation is achieved according to the ship path, which is a straight line or a circle. The related parameters are deduced to obtain the final poses the ship and the vehicle reach simultaneously. Furthermore, the approaching method is summarized. The simulations under different conditions have proved that the proposed method is effective to realize the autonomous approach while meeting real-time performance.

REFERENCES

[1] D. Erdos, A. Erdos, and S. E. Watkins, "An experimental UAV system for search and rescue challenge," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 28, no. 5, pp. 32–37, May 2013.

[2] H. Liu, G. Lu, and Y. Zhong, "Robust LQR attitude control of a 3-DOF laboratory helicopter for aggressive maneuvers," *IEEE Trans. Ind. Electron.*, vol. 60, no. 10, pp. 4627–4636, Oct. 2013.

[3] M. F. Fallon, J. Folkesson, H. McClelland, and J. J. Leonard, "Relocating underwater features autonomously using sonar-based SLAM," *IEEE J. Oceanic Eng.*, vol. 38, no. 3, pp. 500–513, Jul. 2013.

[4] I. Bayezit and B. Fidan, "Distributed cohesive motion control of flight vehicle formations," *IEEE Trans. Ind. Electron.*, vol. 60, no. 12, pp. 5763–5772, Dec. 2013.

[5] Y. Du, J. Fang, and C. Miao, "Frequency domain system identification of an unmanned helicopter based on adaptive genetic algorithm," *IEEE Trans. Ind. Electron.*, vol. 61, no. 2, pp. 870–881, Feb. 2014.

[6] Y. Tang, H. Gao, J. Kurths, and J. Fang, "Evolutionary pinning control and its application in UAV coordination," *IEEE Trans. Ind. Informat.*, vol. 8, no. 4, pp. 828–838, Nov. 2012.

[7] F. Lin, X. Dong, B. M. Chen, K. Y. Lum, and H. L. Tong, "A robust real-time embedded vision system on an unmanned rotorcraft for ground target following," *IEEE Trans. Ind. Electron.*, vol. 59, no. 2, pp. 1038–1049, Feb. 2012.

[8] K. Zhang, E. Collins, and D. Shi, "Centralized and distributed task allocation in multi-robot teams via a stochastic clustering auction," *ACM Trans. Auton. Adapt. Syst.*, vol. 7, no. 2, pp. 1–22, 2012.

[9] K. Zhang, E. Collins, and A. Barbu, "Efficient stochastic clustering auctions for task allocation in homogeneous and heterogeneous teams," *J. Intell. Robot. Syst.*, vol. 72, no. 3, pp. 541–558, Dec. 2013.

[10] G. Ambrosino *et al.*, "Path generation and tracking in 3-D for UAVs," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 4, pp. 980–988, Jul. 2009.

[11] M. Storvik, "Guidance system for automatic approach to a ship," M.S. thesis, Norwegian Univ. Sci. Technol., Trondheim, Norway, 2003.

[12] G. Q. Huang, Y. P. Lu, and Y. Nan, "A survey of numerical algorithms for trajectory optimization of flight vehicles," *Sci. China Technol. Sci.*, vol. 55, no. 9, pp. 2538–2560, Sep. 2012.

[13] K. Yang *et al.*, "Spline-based RRT path planner for non-holonomic robots," *J. Intell. Robot. Syst.*, vol. 73, no. 1–4, pp. 763–782, Jan. 2014.

[14] L. Palmieri and K. O. Arras, "A novel RRT extend function for efficient and smooth mobile robot motion planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Chicago, IL, USA, 2014, pp. 205–211.

[15] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT^*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Chicago, IL, USA, 2014, pp. 2997–3004.

[16] Y. Lu, X. Huo, O. Arslan, and P. Tsiotras, "Incremental multi-scale search algorithm for dynamic path planning with low worst-case complexity," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 6, pp. 1556–1570, Dec. 2011.

[17] S. M. Persson and I. Sharf, "Sampling-based A* algorithm for robot path-planning," *Int. J. Robot. Res.*, vol. 33, no. 13, pp. 1683–1708, 2014.

[18] E. I. Grotli and T. A. Johansen, "Path planning for UAVs under communication constraints using SPLAT! and MILP," *J. Intell. Robot. Syst.*, vol. 65, no. 1–4, pp. 265–282, Jan. 2012.

[19] N. Sudha and A. R. Mohan, "Hardware-efficient image-based robotic path planning in a dynamic environment and its FPGA implementation," *IEEE Trans. Ind. Electron.*, vol. 58, no. 5, pp. 1907–1920, May 2011.

[20] G. E. Jan, C. C. Sun, W. C. Tsai, and T. Lin, "An $O(n \log n)$ shortest path algorithm based on Delaunay triangulation," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 2, pp. 660–666, Apr. 2014.

[21] B. Zhang, L. Tang, J. DeCastro, M. Roemer, and K. Goebel, "A recursive receding horizon planning for unmanned vehicles," *IEEE Trans. Ind. Electron.*, vol. 62, no. 5, pp. 2912–2920, May 2015.

[22] C. C. Tsai, H. C. Huang, and C. K. Chan, "Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation," *IEEE Trans. Ind. Electron.*, vol. 58, no. 10, pp. 4813–4821, Oct. 2011.

[23] C. Cheng, K. Fallahi, H. Leung, and C. K. Tse, "A genetic algorithm-inspired UUV path planner based on dynamic programming," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 1128–1134, Nov. 2010.

[24] T. Berglund, A. Brodnik, H. Jonsson, M. Staffanson, and I. Soderkvist, "Planning smooth and obstacle-avoiding B-spline paths for autonomous mining vehicles," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 1, pp. 167–172, Jul. 2010.

[25] K. Yang and S. Sukkarieh, "An analytical continuous-curvature path-smoothing algorithm," *IEEE Trans. Robot.*, vol. 26, no. 3, pp. 561–568, Jun. 2010.

[26] H. Oh *et al.*, "Coordinated standoff tracking using path shaping for multiple UAVs," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 1, pp. 348–363, Jan. 2014.

[27] F. Ghilardelli, G. Lini, and A. Piazzi, "Path generation using η^4-splines for a truck and trailer vehicle," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 1, pp. 187–203, Jan. 2014.

[28] F. Belkhouche and B. Bendjilali, "Reactive path planning for 3-D autonomous vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 20, no. 1, pp. 249–256, Jan. 2012.

[29] Y. Wang, S. Wang, M. Tan, and Q. Wei, "Real-time dynamic Dubins-Helix method for 3-D trajectory smoothing," *IEEE Trans. Control Syst. Technol.*, 2014, vol. 23, no. 2, pp. 730– 736, Mar. 2015.

[30] S. Hota and D. Ghose, "Optimal path planning for an aerial vehicle in 3-D space," in *Proc. 49th IEEE Conf. Decis. Control*, Atlanta, GA, USA, 2010, pp. 4902–4907.

Yu Wang received the B.S. degree in automation from Beijing Institute of Technology, Beijing, China, in July 2011. He is currently working toward the Ph.D. degree in control theory and control engineering in the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing.

He is currently with the University of Chinese Academy of Sciences, Beijing. His research interests include intelligent control, robotics, and biomimetic robots.

Shuo Wang received the B.E. degree in electrical engineering from Shenyang Architectural and Civil Engineering Institute, Shenyang, China, in 1995, the M.E. degree in industrial automation from Northeastern University, Shenyang, in 1998, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2001.

He is currently a Professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. His research interests include biomimetic robot, underwater robot, and multirobot systems.

Min Tan received the B.E. degree from Tsinghua University, Beijing, China, in 1986 and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 1990.

He is currently a Professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. His research interests include advanced robot control, biomimetic robot, and multirobot system.