

Research Article

Fast Path Planning for On-Water Automatic Rescue Intelligent Robot Based on Constant Thrust Artificial Fluid Method

Yongqiang Qi  and Yi Ke

School of Mathematics, China University of Mining and Technology, Xuzhou, Jiangsu 221116, China

Correspondence should be addressed to Yongqiang Qi; qiyongqiang3@163.com

Received 8 August 2020; Revised 3 September 2020; Accepted 3 October 2020; Published 21 October 2020

Academic Editor: Cristian Mateos

Copyright © 2020 Yongqiang Qi and Yi Ke. 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.

In this paper, fast path planning of on-water automatic rescue intelligent robot is studied based on the constant thrust artificial fluid method. First, a three-dimensional environment model is established, and then the kinematics equation of the robot is given. The constant thrust artificial fluid method based on the isochronous interpolation method is proposed, and a novel algorithm of constant thrust fitting is researched through the impulse compensation. The effect of obstacles on original fluid field is quantified by the perturbation matrix, and the streamlines can be regarded as the planned path. Simulation results demonstrate the effectiveness of this method by comparing with A-star algorithm and ant colony algorithm. It is proved that the planned path can avoid all obstacles smoothly and swiftly and reach the destination eventually.

1. Introduction

Robot path planning is based on a certain performance index (the shortest path, the least cost, the least time, etc.) to find the best path from the starting point to the target point that can avoid obstacles. At present, robot path planning mainly includes artificial intelligence path planning technology, ant colony algorithm, A-star algorithm, artificial potential field path planning technology and map construction path planning technology, etc.

The on-water automatic rescue intelligent robot is composed of four parts: monitoring device, unmanned automatic search and rescue boat, lifebuoy, and information receiving device as shown in Figure 1. It integrates the functions of falling water detection, falling water warning, rescue, GPS positioning, image transmission, information collection, and so on. When someone enters the dangerous area, the on-water intelligent search and rescue robot will sound an alarm to prevent accidents. If it detects that someone has fallen into the water, it will calculate the position and distance through infrared and sound sensing technology, ultrasonic distance sensors, etc., and then quickly send the lifebuoy to the drowning person so that he

can save himself. Therefore, the on-water automatic rescue intelligent robot can carry out rescue more efficiently and it is of great significance to study the fast path planning of on-water automatic rescue intelligent robots.

Fast path planning is very important to realize autonomous search and rescue of water robots. Fast path planning generally includes indirect path planning and direct path planning [1]. Indirect path planning is divided into outer loop guidance and inner loop control. Indirect path planning has a simple structure and is widely used in engineering [2]. It mainly includes backstepping, internal model control, fuzzy control, neural network, and so on. Among them, the backstepping method and internal model control can get a better planning path, but an accurate mathematical model of the controlled object is required [3]. Fuzzy control and neural networks do not require precise controlled object models but require designers to have rich knowledge and practical experience, so the error fluctuates greatly. The direct path planning is to integrate the two links, usually based on neural networks and other methods to achieve fast path planning [4].

The S-plane control method is a combination of fuzzy control and PID control. It has the advantages of not relying



FIGURE 1: On-water automatic rescue intelligent robot.

on precise mathematical models, suitable for nonlinear control, and easy to adjust parameters. The S-plane control method is an effective and simple method for the motion control of underwater robots, and the experiment verifies that the S-plane controller has both fast and stable control effects [5]. For the research of outer loop guidance, LOS (line-of-sight) tracking has been widely used in wheeled robots, aircraft, and surface ships. Fossen et al. added an integral link to the traditional LOS guidance algorithm to eliminate the lateral deviation of the ship's position caused by the slowly changing environment and achieved good control effects [6].

In addition, the problem of robot path planning in an unknown environment has attracted more and more researchers' attention [7]. The main research methods include the genetic algorithm, the gradient algorithm, the nonlinear mapping algorithm, the linear mapping algorithm, the reinforcement learning algorithm, and so on [8]. The artificial potential field (APF) method has the advantages of simple principle and real-time computation [9], but there exists local minimum when the robot enters into a concave area. Besides, it is hard to obtain a feasible path sometimes even if the magnitude of attractive or repulsive force is regulated. The intelligent algorithms, such as particle swarm optimization (PSO) [10], evolutionary algorithm (EA) [11], and ant colony algorithm (ACO) [12], are also widely used to path planning. These methods can be easily employed in different environments, but it is possible to trap in a local optimum. However, the aforementioned drawback can be relieved when the intelligent methods are improved or combined with other methods [13].

These traditional approaches are improved to solve the path planning problem. However, the calculation of these algorithms tends to increase exponentially if the planning space enlarges [14]. Besides, the planned path may be not smooth enough for robot to track. As a result, extra strategy of smoothing path is usually needed [15]. A novel algorithm based on disturbed fluid and trajectory propagation is developed to solve the path planning problem of unmanned aerial vehicle in static environment [16]. The core path graph (CPG) algorithm [17] calculates the CPG where arcs are minimum-length trajectories satisfying geometrical constraints and searches the optimal trajectory between two arbitrary nodes of the graph. In addition, the conclusions about robot maneuvering such as solutions to the characteristic equation for industrial robot's elliptic trajectories

[18], experimental investigations of a highly maneuverable mobile omniwheel robot [19], navigation control and stability investigation of a mobile robot based on a hexacopter equipped with an integrated manipulator [20], and integration of inertial sensor data into control of the mobile platform [21] have also been studied in recent years, providing important theoretical and experimental reference values for this paper.

This study aims at constant thrust maneuver of on-water automatic rescue intelligent robot. The main contributions of this study are as follows: (1) the 3-D environment model is established by using the Taylor formula of multivariate functions; (2) a novel algorithm of constant thrust fitting is proposed through the impulse compensation and the isochronous interpolation method; and (3) the tangential vector and disturbance matrix of the artificial fluid method are improved by combing the interfered fluid dynamical system.

The rest of the paper is organized as follows. The 3-D environmental model is established in Section 2. Section 3 focuses on the kinematics model of on-water automatic rescue intelligent robot. The constant thrust artificial fluid method is given in Section 4. The simulation studies and results are presented in Section 5, followed by the conclusions and future work in Section 6.

2. Establishing 3-D Environmental Model

The purpose of on-water automatic rescue intelligent robot fast path planning is to avoid obstacles and reach destination. In this paper, obstacles are described approximately as 3-D space surfaces when building the 3-D environment model. The relative motion coordinate system $o-xyz$ is defined as the path planning space, and $R = (x, y, z)$ is defined as the position of the robot relative to the obstacle. Suppose that the parametric equations of the obstacle's space curved surface are as follows:

$$f(u, v) = \begin{cases} x = x(u, v), \\ y = y(u, v), \\ z = z(u, v), \end{cases} \quad (1)$$

where $u \in [a, b]$ and $v \in [c, d]$. Assuming that $Q(u, v)$ is any point on the parametric surface (1), $R = (x, y, z)$ is transformed as $R(u, v)$, and then the normal vector at point $Q(u, v)$ is

$$n_q = f_u \times f_v, \quad (2)$$

where f_u and f_v are the partial derivative of $f(u, v)$ on u and v . Obviously, in order to get the minimum distance between $R(u, v)$ and $Q(u, v)$, RQ should be parallel to the normal vector n_q :

$$RQ \times n_q = 0. \quad (3)$$

Equation (3) can be written as follows:

$$\begin{cases} f_1(u, v) = 0, \\ f_2(u, v) = 0, \\ f_3(u, v) = 0, \end{cases} \quad (4)$$

where f_1, f_2, f_3 are nonlinear equations. Then, equation (4) can be written as follows:

$$F(u, v) = \begin{pmatrix} f_1(u, v) \\ f_2(u, v) \\ f_3(u, v) \end{pmatrix} = 0. \quad (5)$$

The derivative matrix of equation (5) is as follows:

$$F'(u, v) = \begin{pmatrix} \frac{\partial f_1(u, v)}{\partial u} & \frac{\partial f_1(u, v)}{\partial v} \\ \frac{\partial f_2(u, v)}{\partial u} & \frac{\partial f_2(u, v)}{\partial v} \\ \frac{\partial f_3(u, v)}{\partial u} & \frac{\partial f_3(u, v)}{\partial v} \end{pmatrix}. \quad (6)$$

Assume that $X^*(u^*, v^*)$ is the optimal solution of equation (5); then, the following results can be obtained by using the Taylor formula of multivariate functions.

$$f_i(X^*) + \left(\Delta u \frac{\partial}{\partial u} + \Delta v \frac{\partial}{\partial v} \right) f_i(X^*) = 0, \quad (7)$$

where $i = 1, 2, 3$, $\Delta u = u - u^*$, and $\Delta v = v - v^*$. X_k is the solution of equation (7) and can be seen as an approximate solution of X^* . Equation (7) can be transformed into a matrix equation:

$$F(X^*) + F'(X^*)(X_k - X^*) = 0. \quad (8)$$

Then, the least squares solution of X^* can be obtained:

$$X^* = X_k + \left[(F'(X^*))^T F'(X^*) \right]^{-1} (F'(X^*))^T F(X^*). \quad (9)$$

Therefore, the shortest distance between the robot and the obstacle is $D^*(u, v) = |RQ^*|$, where $|\cdot|$ represents the modular of vector from $R(u, v)$ to $Q^*(u, v)$. Then, the 3-D environment model is established using the shortest distance theory, and the specific model will be given in the simulation example.

3. Kinematics Model of On-Water Automatic Rescue Intelligent Robot

In the process of fast path planning of the on-water automatic rescue intelligent robot, it is necessary to consider the three directions of motion laws of longitudinal, lateral, and yaw. The motion state of the on-water automatic rescue intelligent robot is represented by $\eta_1 = [x, y, \varphi]^T$ in the earth coordinate system, the state of three degrees of freedom is represented by $\eta_2 = [u, v, r]^T$ in the body coordinate system, and the control input is represented by $\{T_1, T_2, T_3\}$. Then, the kinematics model is expressed as follows:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{pmatrix} = \begin{pmatrix} \cos \varphi & \sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ r \end{pmatrix}. \quad (10)$$

In order to simplify the calculation, when the dynamic modeling of the on-water automatic rescue intelligent robot is established, the resistance of the robot is considered to be linear with its speed, and the dynamic equation of the robot is as follows:

$$M\dot{v} + (C + D)v = \tau, \quad (11)$$

where M is the inertial mass matrix of the robot, C is the coefficient matrix of centripetal force and Coriolis force, D is the resistance coefficient matrix, and τ is the thrust and moment matrix of the robot. The coefficient in the formula is obtained through actual test. Therefore, the dynamic model of the robot is as follows:

$$\begin{cases} \dot{u} = \frac{m_{22}}{m_{11}} v r - \frac{d_{11}}{m_{11}} u + \frac{\tau_1}{m_{11}}, \\ \dot{v} = \frac{m_{11}}{m_{22}} u r - \frac{d_{22}}{m_{22}} v, \\ \dot{r} = \frac{m_{11} - m_{22}}{m_{33}} u v - \frac{d_{33}}{m_{33}} r + \frac{\tau_3}{m_{33}}, \end{cases} \quad (12)$$

where m_{ii} and d_{ii} are the main diagonal elements of the M and D . The resultant force vectors in the three directions generated by the robot's thrusters are as follows:

$$\begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & l & -l \end{pmatrix} \begin{pmatrix} T_1 \\ T_2 \\ T_3 \end{pmatrix}. \quad (13)$$

where l is the distance from the thruster to the centroid of the robot, $T_d = [T_x, T_y, T_z]^T$ is the expected output force, $\tau = [T_1, T_2, T_3]^T$ is thrust output of the thruster, and B is the placement matrix of the thruster.

It is worth noting that because the placement matrix of the thruster is a non-full rank matrix, it cannot be calculated by the inverse of the matrix. The traditional method is to use the pseudoinverse matrix to distribute the thrust, but this method does not consider the constraint factors of the thrust amplitude, which is easy to cause the oversaturation of the thrust distribution. Therefore, in this section, the quadratic programming method is used to introduce performance indicators to establish three thrust distribution functions as follows:

$$\begin{aligned} \min J &= (B\tau - T_d)^T (B\tau - T_d), \\ \text{s.t.}, \tau_{\min} &\leq \tau \leq \tau_{\max}. \end{aligned} \quad (14)$$

4. Fast Path Planning Based on Constant Thrust Artificial Fluid Method

4.1. Constant Thrust Collision Avoidance Maneuver. In the process of path planning, collision avoidance maneuvers should be considered. Suppose that the time of on-water automatic rescue intelligent robot's collision avoidance maneuver is T and the shortest switching time interval is ΔT .

There are M shortest switching time intervals and N target maneuver positions. T_i represents the time of the i -th thrust arc, and M_i denotes the number of shortest switching time intervals in the i -th thrust arc.

$$\begin{aligned} T &= M\Delta T, \\ T_i &= M_i\Delta T, \\ M &= \sum_{j=1}^M M_j. \end{aligned} \quad (15)$$

Suppose that the actual constant thrusts of the robot is F , the maximum thrusts is \widehat{F} , and the theoretical continuous thrusts is F^* . The thrusters can provide different sizes of constant thrust to meet different thrust requirements. There are N different sizes of constant thrust which can be denoted as follows:

$$\frac{\widehat{F}}{N}, \frac{2\widehat{F}}{N}, \frac{3\widehat{F}}{N}, \dots, \frac{(N-1)\widehat{F}}{N}, \widehat{F}. \quad (16)$$

The size of the constant thrust is calculated as follows. There are $(N+1)$ thrust levels that can be selected: $(L\widehat{F}/N)$, $L = (0, 1, 2, \dots, N)$, and the level of the constant thrust can be calculated as follows, taking the i -th thrust arc as example:

$$L = \left[\frac{N \int_{T_i}^{T_i+\Delta T} |F^*(t)| dt}{\widehat{F}\Delta T} \right], \quad (17)$$

where $[\]$ represents the bracket function and $|F^*(t)|$ represents the absolute value of $F^*(t)$.

The constant thrust fitting should be discussed in several categories, for convenience, taking the i -th thrust arc as example. Suppose that the theoretical working time in the i -th thrust arc is t^* . The simplest case is the theoretical working time in the i -th thrust arc $t^* = 0$; then, the actual constant thrust is $F = F^* = 0$. If the theoretical working time in the i -th thrust arc is $(\Delta T \leq t^* \leq T_i = M_i\Delta T)$, then the constant thrust fitting should be discussed in the following situations.

Case 1. $t^* = \Delta T < T_i$.

Without loss of generality, assume that ΔI_i is the impulse error in the i -th thrust arc and can be calculated as follows:

$$\Delta I_i = \text{sgn}(F^*(t)) \left| \int_{T_i}^{T_i+\Delta T} |F^*(t)| dt - \frac{L\widehat{F}\Delta T}{N} \right|. \quad (18)$$

Suppose that the value of the impulse compensation threshold is a positive constant $\gamma > 0$. When ΔI_i satisfies the following condition:

$$\left| \int_{T_i}^{T_i+\Delta T} |F^*(t)| dt - \frac{\widehat{F}\Delta T}{N} \left[\frac{N \int_{T_i}^{T_i+\Delta T} |F^*(t)| dt}{\widehat{F}\Delta T} \right] \right| \leq \gamma, \quad (19)$$

the actual constant thrust of the robot can be calculated as follows:

$$F = \text{sgn}(F^*(t)) \frac{\widehat{F}\Delta T}{N} \left[\frac{N \int_{T_i}^{T_i+\Delta T} |F^*(t)| dt}{\widehat{F}\Delta T} \right]. \quad (20)$$

Then, the robot does not need to carry out impulse compensation. When the impulse error ΔI_i satisfies the following condition:

$$\gamma < \left| \int_{T_i}^{T_i+\Delta T} |F^*(t)| dt - \frac{\widehat{F}\Delta T}{N} \left[\frac{N \int_{T_i}^{T_i+\Delta T} |F^*(t)| dt}{\widehat{F}\Delta T} \right] \right|, \quad (21)$$

then the robot should carry out impulse compensation and the size of the constant thrust impulse compensation can be calculated as follows:

$$\Delta I_i = \text{sgn}(F^*(t)) \frac{\widehat{F}\Delta T}{N}. \quad (22)$$

Case 2. $t^* = M_i\Delta T$.

For this situation, the impulse error in the i -th thrust arc ΔI_i can be calculated as follows:

$$\Delta I_i = \sum_{j=0}^{M_i} \int_{T_i+j\Delta T}^{T_i+(j+1)\Delta T} \left\{ F^*(t) - \text{sgn}(F^*(t)) \frac{\widehat{F}}{N} \left[\frac{N \int_{T_i+j\Delta T}^{T_i+(j+1)\Delta T} |F^*(t)| dt}{\widehat{F}\Delta T} \right] \right\} dt. \quad (23)$$

If there exist n_1 shortest switching time intervals satisfying the following conditions, without loss of generality, suppose that these time intervals are the first n_1 shortest switching time interval, taking the j -th shortest switching time interval as an example:

$$\left| \int_{T_i+j\Delta T}^{T_i+(j+1)\Delta T} F^*(t) dt \right| \leq \gamma. \quad (24)$$

Then, the size of the impulse compensation can be calculated as follows. When the impulse error ΔI_i satisfies the following condition:

$$\left| \sum_{j=0}^{M_i} \int_{T_i+j\Delta T}^{T_i+(j+1)\Delta T} \left\{ F^*(t) - \operatorname{sgn}(F^*(t)) \frac{\widehat{F}}{N} \left[\frac{N \int_{T_i+j\Delta T}^{T_i+(j+1)\Delta T} |F^*(t)| dt}{\widehat{F}\Delta T} \right] \right\} dt \right| \leq \gamma, \quad (25)$$

the actual constant thrust of the robot can be calculated as follows, taking the j -th shortest switching time interval as an example:

$$F = \int_{T_i+j\Delta T}^{T_i+(j+1)\Delta T} \operatorname{sgn}(F^*(t)) dt \frac{\widehat{F}}{N} \frac{N \int_{T_i+j\Delta T}^{T_i+(j+1)\Delta T} |F^*(t)| dt}{\widehat{F}\Delta T}, \quad j = \{0, 1, \dots, M_i\}. \quad (26)$$

Then, the robot will not carry out impulse compensation.

When the impulse error ΔI_{zi} satisfies the following condition:

$$\left| \int_{T_i}^{T_i+M_i\Delta T} F^*(t) dt - \int_{T_i+(n_1+1)\Delta T}^{T_i+M_i\Delta T} \operatorname{sgn}(F^*(t)) \frac{\widehat{F}}{N_z} \left[\frac{N \int_{T_i+j\Delta T}^{T_i+(j+1)\Delta T} |F^*(t)| dt}{\widehat{F}\Delta T} \right] dt \right| \leq \gamma, \quad (27)$$

and there is a constant n_2 that satisfies the following equation:

$$\left[\int_{T_i}^{T_i+M_i\Delta T} \frac{F^*(t) N_z}{\widehat{F}\Delta T} dt - \int_{T_i+(n_1+1)\Delta T}^{T_i+M_i\Delta T} \frac{\operatorname{sgn}(F^*(t))}{\Delta T} \left[\frac{N \int_{T_i+j\Delta T}^{T_i+(j+1)\Delta T} |F^*(t)| dt}{\widehat{F}\Delta T} \right] dt \right] = n_2, \quad (28)$$

the actual constant thrust of the robot can be calculated as follows:

$$\int_{T_i+(n_1+1)\Delta T}^{T_i+M_i\Delta T} \operatorname{sgn}(F^*(t)) \frac{\widehat{F}}{N} \left[\frac{N \int_{T_i+j\Delta T}^{T_i+(j+1)\Delta T} |F^*(t)| dt}{\widehat{F}\Delta T} \right] dt. \quad (29)$$

Then, the robot does not need to carry out impulse compensation. When the impulse error ΔI_{zi} satisfies the following condition:

$$\left| \int_{T_i}^{T_i+M_i\Delta T} F^*(t) dt - \int_{T_i+(n_1+1)\Delta T}^{T_i+M_i\Delta T} \operatorname{sgn}(F^*(t)) \frac{\widehat{F}}{\Delta T} \left[\frac{N \int_{T_i+j\Delta T}^{T_i+(j+1)\Delta T} |F^*(t)| dt}{\widehat{F}\Delta T} \right] dt \right| > \gamma, \quad (30)$$

then the robot should carry out impulse compensation and the size of the constant thrust impulse compensation can be calculated as follows:

$$\Delta I_i = n_2 \operatorname{sgn}(F^*(t)) \Delta T. \quad (31)$$

Case 3. $t^* = M_1 \Delta T$ ($1 < m_1 < M_i$).

Without loss of generality, suppose that t^* is the first m_1 shortest switching time interval and the impulse error in the i -th thrust arc ΔI_i can be calculated as follows:

$$\Delta I_i = \sum_{j=0}^{M_1} \int_{T_i+j\Delta T}^{T_i+(j+1)\Delta T} \left\{ F^*(t) - \operatorname{sgn}(F^*(t)) \frac{\widehat{F}}{N} \left[\frac{N \int_{T_i+j\Delta T}^{T_i+(j+1)\Delta T} |F^*(t)| dt}{\widehat{F}\Delta T} \right] \right\} dt. \quad (32)$$

Furthermore, if there are m_1 shortest switching time intervals satisfying the following conditions, without loss of generality, suppose that these time intervals are the top m_1 shortest switching time interval, taking the j -th shortest switching time interval as an example:

$$\left| \int_{T_i+j\Delta T}^{T_i+(j+1)\Delta T} F^*(t) dt \right| \leq \gamma. \quad (33)$$

Then, the size of the impulse compensation can be calculated as follows. When ΔI_i satisfies the following condition:

$$\left| \sum_{j=0}^{M_1} \int_{T_i+j\Delta T}^{T_i+(j+1)\Delta T} \left\{ F^*(t) - \operatorname{sgn}(F^*(t)) \frac{\widehat{F}}{N} \left[\frac{N \int_{T_i+j\Delta T}^{T_i+(j+1)\Delta T} |F^*(t)| dt}{\widehat{F}\Delta T} \right] \right\} dt \right| \leq \gamma, \quad (34)$$

the actual constant thrust of the robot can be calculated as follows, taking the j -th shortest switching time interval as an example:

$$F = \int_{T_i+j\Delta T}^{T_i+(j+1)\Delta T} \operatorname{sgn}(F^*(t)) dt \frac{\widehat{F}}{N} \left[\frac{N \int_{T_i+j\Delta T}^{T_i+(j+1)\Delta T} |F^*(t)| dt}{\widehat{F}\Delta T} \right]. \quad (35)$$

Then, the robot will not carry out impulse compensation. When the impulse error ΔI_i satisfies the following condition:

$$\left| \int_{T_i}^{T_i+M_1\Delta T} F^*(t) dt - \int_{T_i+(m_1+1)\Delta T}^{T_i+M_1\Delta T} \operatorname{sgn}(F^*(t)) \frac{\widehat{F}}{N} \left[\frac{N \int_{T_i+j\Delta T}^{T_i+(j+1)\Delta T} |F^*(t)| dt}{\widehat{F}\Delta T} \right] dt \right| \leq \gamma, \quad (36)$$

and there is a constant m_2 that satisfies the following equation:

$$\left[\left| \int_{T_i}^{T_i+M_1\Delta T} \frac{F^*(t)N}{\widehat{F}\Delta T} dt - \int_{T_i+(m_1+1)\Delta T}^{T_i+M_1\Delta T} \frac{\operatorname{sgn}(F^*(t))}{\Delta T} \left[\frac{N \int_{T_i+j\Delta T}^{T_i+(j+1)\Delta T} |F^*(t)| dt}{\widehat{F}\Delta T} \right] dt \right| \right] = m_2, \quad (37)$$

then the actual constant thrust of the robot can be calculated as follows:

$$F = \int_{T_i+(m_1+1)\Delta T}^{T_i+M_1\Delta T} \operatorname{sgn}(F^*(t)) dt \frac{\widehat{F}}{N} \left[\frac{N \int_{T_i+j\Delta T}^{T_i+(j+1)\Delta T} |F^*(t)| dt}{\widehat{F}\Delta T} \right]. \quad (38)$$

The robot will not carry out impulse compensation. When the impulse error ΔI_i satisfies the following condition:

$$\left| \int_{T_i}^{T_i+M_1\Delta T} F^*(t) dt - \int_{T_i+(m_1+1)\Delta T}^{T_i+M_1\Delta T} \operatorname{sgn}(F^*(t)) \frac{\widehat{F}}{N} \left[\frac{N \int_{T_i+j\Delta T}^{T_i+(j+1)\Delta T} |F^*(t)| dt}{\widehat{F}\Delta T} \right] dt \right| > \gamma, \quad (39)$$

then the robot should carry out impulse compensation and the constant thrust impulse compensation can be calculated as follows:

$$\Delta I_i = m_2 \text{sgn}(F^*(t)) \Delta T. \quad (40)$$

4.2. Path Planning Based on the Artificial Fluid Method. Based on the description in Section 3, the procedure for path planning is as follows. First, the improved perturbation matrix $P(u, v)$ is calculated. Next, the disturbed fluid velocity v_d is calculated by modifying the target velocity v_T . Then, the planned path is obtained by the recursive integration of v_d . Finally, the switching control laws based on the isochronous interpolation method are given. To describe the influence of obstacle on the original flow, the improved perturbation matrix $P(u, v)$ is defined as follows [16]:

$$P(u, v) = I - \left(\frac{n_q n_q^T}{D^*(u, v)^{\rho(u, v)} \|n_q\|^2} \right) + \left(\frac{\lambda(u, v) f_u n_q^T}{D^*(u, v)^{\sigma(u, v)} \|f_u\| \|n_q\|} \right), \quad (41)$$

where I is a 2×2 identity matrix, n_q is a column vector, f_u is a tangential vector (the partial derivative of $f(u, v)$ on u) at the point $Q(u, v)$, $\lambda(u, v)$ is a saturation function defining the orientation of tangential velocity, and $\|\cdot\|$ is 2-norm of a vector or a matrix. Besides, $\rho(u, v)$ and $\sigma(u, v)$ are defined as the weight of n_q and t_q , respectively.

$$\rho(u, v) = \rho_0 \exp\left(\frac{1}{D_0 - D^*(u, v)}\right),$$

$$\begin{aligned} D^*(u, v)^{\rho(u, v)} &= (D_0 + \delta(u, v))^{\rho_0 e^{-(1/\delta(u, v))}}, \\ &= D_0 \rho_0 e^{-(1/\delta(u, v))} \left(\left(1 + \frac{\delta(u, v)}{D_0} \right)^{(D_0/\delta(u, v))} \right)^{(\delta(u, v)/D_0) \rho_0 e^{-(1/\delta(u, v))}}. \end{aligned} \quad (44)$$

It can be inferred that $\delta(u, v) \rightarrow 0+$; then, $D_0 \rho_0 e^{-(1/\delta(u, v))} \rightarrow 0$, $(1 + (\delta(u, v)/D_0))^{(D_0/\delta(u, v))} \rightarrow e$, and $(\delta(u, v)/D_0) \rho_0 e^{-(1/\delta(u, v))} \rightarrow 0$; therefore, $D^*(u, v)^{\rho(u, v)} \rightarrow 1$.

$$P(u, v) = I - \frac{n_q n_q^T}{\|n_q\|^2} + \frac{\lambda(u, v) f_u n_q^T}{\|f_u\| \|n_q\|}. \quad (45)$$

Therefore,

$$n_q^T v_d = n_q^T \left(I - \frac{n_q n_q^T}{\|n_q\|^2} + \frac{\lambda(u, v) f_u n_q^T}{\|f_u\| \|n_q\|} \right) v_T. \quad (46)$$

Because $n_q^T f_u = 0$ means that $n_q^T v_d = 0$, the path is outside of the minimum permitted distance and there is no collision.

$$\begin{aligned} \sigma(u, v) &= \sigma_0 \exp\left(\frac{1}{D_0 - D^*(u, v)}\right), \\ \lambda(u, v) &= \begin{cases} 1, & v_T^T f_u n_q^T v_T > \lambda_0, \\ \frac{v_T^T f_u n_q^T v_T}{\lambda_0}, & -\lambda_0 \leq v_T^T f_u n_q^T v_T \leq \lambda_0, \\ -1, & v_T^T f_u n_q^T v_T < -\lambda_0, \end{cases} \end{aligned} \quad (42)$$

where ρ_0 is the repulsive parameter, σ_0 is the tangential parameter, D_0 is the minimum permitted distance between the boundary of on-water automatic rescue intelligent robot, and λ_0 is a small positive threshold of the saturation function $\lambda(u, v)$. Then, the disturbed fluid velocity v_d can be calculated by

$$v_d = P(u, v) v_T. \quad (43)$$

4.3. The Planned Path Can Avoid Obstacles Safely

Proof. Vectors n_q and f_u are perpendicular exactly: $n_q^T f_u = 0$. Suppose that the distance between the boundary of on-water automatic rescue intelligent robot is close to the minimum permitted distance D_0 . Suppose that $D^* = D_0 + \delta(u, v)$ is a monotonically decreasing function.

4.4. The Planned Path Can Reach the Destination Eventually

Proof. Because the goal of the path planning is to make the on-water automatic rescue intelligent robot reach the destination safely, the velocity of the on-water automatic rescue intelligent robot should have a component in the direction of the target velocity, i.e., velocities v_T and v_d should satisfy $v_T^T v_d \geq 0$, and the planned path will converge to the target point. Besides, the condition $v_d \approx v_T$ should be satisfied when the on-water automatic rescue intelligent robot is near to the destination $\xi_T = (x_T, y_T, z_T)$.

$$v_T^T v_d = \|v_T\|^2 \left(1 - \frac{\cos^2 \langle v_T, n_q \rangle}{D^*(u, v)^{\rho(u, v)}} \right) + \frac{\lambda(u, v) v_T^T f_u n_q^T v_T}{D^*(u, v)^{\sigma(u, v)} \|f_u\| \|n_q\|}, \quad (47)$$

where (\cdot) is inner product of vectors and $\langle v_T, n_q \rangle$ denotes the angle between v_T and n_q . It is obvious that when the on-water automatic rescue intelligent robot approaches the destination, then $D^*(u, v) \rightarrow 0+$ and $(D^*(u, v))^{\rho_0 \exp(1/D_0 - D^*(u, v))} \rightarrow +\infty$. Furthermore,

$1 - ((\cos^2 \langle v_T, n_q \rangle) / (D^*(u, v)^{\rho(u, v)})) \geq 0$ and $v_T^T v_d \geq 0$ hold. When the robot approaches the destination, $D^*(u, v)^{\rho(u, v)} \rightarrow +\infty$ and $(D^*(u, v))^{\sigma(u, v)} \rightarrow +\infty$. It can be inferred that $P(u, v) \rightarrow I$ and $v_d \approx v_T$ hold.

4.5. Analysis of the Disturbed Fluid Velocity v_d . The modified velocity v_d defined by equation (43) can be expressed as

$$v_d = v_T - \frac{n_q^T v_T}{D^*(u, v)^{\rho(u, v)} \|n_q\|^2} n_q + \frac{\lambda(u, v) n_q^T v_T}{D^*(u, v)^{\sigma(u, v)} \|f_u\| \|n_q\|} f_u. \quad (48)$$

It can be seen that v_d consists of three parts: v_T can be called the target velocity; $(n_q^T v_T / (D^*(u, v)^{\rho(u, v)} \|n_q\|^2)) n_q$ is taken as the repulsive velocity; and $((\lambda(u, v) n_q^T v_T) / (D^*(u, v)^{\sigma(u, v)} \|f_u\| \|n_q\|)) f_u$ can be called the tangential velocity. Similarly, the perturbation matrix $P(u, v)$ can be divided into three parts: attractive matrix I , repulsive matrix $(n_q n_q^T / (D^*(u, v)^{\rho(u, v)} \|n_q\|^2))$, and tangential matrix $(\lambda(u, v) f_u n_q^T / (D^*(u, v)^{\sigma(u, v)} \|f_u\| \|n_q\|))$. It can be analyzed that the magnitudes of repulsive and tangential velocities increase with $\rho(u, v)$ and $\sigma(u, v)$, respectively. Therefore, the shape of the path can be readjusted by changing parameters $\rho(u, v)$ and $\sigma(u, v)$. This method is similar to the virtual force method or the artificial potential field method to some degree. However, the perturbation matrix by this method can describe the effect of obstacles on path more objectively, considering the shape of obstacles and the position of on-water automatic rescue intelligent robot.

4.6. Improved Tangential Vector. The normal vector $n(\xi)$ and the tangential plane which is perpendicular to the normal vector can be accurately obtained. The smaller the angle between the tangential vector $t(\xi)$ and the original speed $v(\xi)$, the greater the component of the tangential vector $t(\xi)$ in the direction of the original velocity $v(\xi)$. Then, the robot can reach the destination faster and make the path shorter. Thus, the tangential vector $t_{\text{new}}(\xi)$ is redefined as follows:

$$t_{\text{new}}(\xi) = \begin{cases} \arg \min_{t(\xi) \cdot n(\xi) = 0} \{\cos \langle v(\xi), t(\xi) \rangle\}, & v(\xi) \neq \omega \cdot n(\xi), \\ \left(\frac{\partial F(\xi)}{\partial y}, -\frac{\partial F(\xi)}{\partial x}, 0 \right)^T, & v(\xi) = \omega \cdot n(\xi), \end{cases} \quad (49)$$

where $\langle v(\xi), t(\xi) \rangle$ refers to the angle between original velocity $v(\xi)$ and vector $t(\xi)$ and ω is a constant.

- (1) When $v(\xi) = \omega \cdot n(\xi)$, the original speed $v(\xi)$ is parallel to normal vector $n(\xi)$. In this case, any direction on the tangential plane is the shortest

direction, and the tangential vector $t(\xi)$ is selected in the horizontal direction in this paper.

- (2) When $v(\xi) \neq \omega \cdot n(\xi)$, the direction of new tangential vector $t_{\text{new}}(\xi)$ is the vertical projection of the original velocity $v(\xi)$ on the tangential plane.

Suppose that original speed $v(\xi)$ and normal vector $n(\xi)$ are defined as follows:

$$v(\xi) = \left(C \frac{x - x_d}{d(\xi)}, C \frac{y - y_d}{d(\xi)}, C \frac{z - z_d}{d(\xi)} \right)^T = (v_1, v_2, v_3)^T, \\ n(\xi) = \left(\frac{\partial F(\xi)}{\partial x}, \frac{\partial F(\xi)}{\partial y}, \frac{\partial F(\xi)}{\partial z} \right)^T = (n_1, n_2, n_3)^T. \quad (50)$$

Then, the normal vector of tangential plane α is $n(\xi)$. Original speed $v(\xi)$ and normal vector $n(\xi)$ are in the cut plane β ; thus, the normal vector $n_\beta(\xi)$ of cut plane β can be obtained:

$$n_\beta(\xi) = v(\xi) \times n(\xi) = \begin{pmatrix} v_2 & v_3 \\ n_2 & n_3 \end{pmatrix} \times \begin{pmatrix} v_3 & v_1 \\ n_3 & n_1 \end{pmatrix} \times \begin{pmatrix} v_1 & v_2 \\ n_1 & n_2 \end{pmatrix} \\ = (v_2 n_3 - v_3 n_2, v_3 n_1 - v_1 n_3, v_1 n_2 - v_2 n_1)^T \\ = (n_{\beta 1}, n_{\beta 2}, n_{\beta 3})^T. \quad (51)$$

Then, $t_{\text{new}}(\xi)$ can be calculated:

$$t_{\text{new}}(\xi) = n_\beta(\xi) \times n(\xi) = \begin{pmatrix} n_{\beta 2} & n_{\beta 3} \\ n_2 & n_3 \end{pmatrix} \times \begin{pmatrix} n_{\beta 3} & n_{\beta 1} \\ n_3 & n_1 \end{pmatrix} \times \begin{pmatrix} n_{\beta 1} & n_{\beta 2} \\ n_1 & n_2 \end{pmatrix}. \quad (52)$$

Furthermore, the improved tangential vector $T(\xi)$ is defined as follows:

$$T(\xi) = \lambda_1 \frac{t(\xi)}{\|t(\xi)\|} + \lambda_2 \frac{t_{\text{new}}(\xi)}{\|t_{\text{new}}(\xi)\|}, \quad (53)$$

where $t(\xi) = ((\partial F(\xi)/\partial y), (\partial F(\xi)/\partial x), 0)^T$, λ_1, λ_2 are inspiration factors, which are determined by the robot properties, and (λ_1/λ_2) is the proportion factor.

4.7. Designing Constant Thrust Switching Control Laws. The constant thrust switching control laws are obtained through the isochronous interpolation method. Without loss of generality, taking case 3 as an example, if the impulse error ΔI_i satisfies the condition of equation (34), the total number of the accelerating time intervals and the decelerating time intervals is M_1 and the number of zero-thrust time intervals is $M_i - M_1$. The position of the three types of time intervals is decided by the curve of the theoretical continuous thrust $F^*(t)$.

If the impulse error ΔI_i satisfies the condition of equation (36), the total number of the accelerating time intervals and the decelerating time intervals is $M_i - M_1$ and the number of zero-thrust time intervals is $M_i - M_1 + m_1$.

The position of the three types of time intervals is decided by the curve of the theoretical continuous thrust $F^*(t)$.

If the impulse error ΔI_i satisfies the condition of equation (39), the total number of the accelerating time intervals and the decelerating time intervals is $M_1 - m_1 + m_2$ and the number of zero-thrust time intervals is $M_i - M_1 + m_1 - m_2$. The position of the three types of time intervals is decided by the curve of the theoretical continuous thrust $F^*(t)$.

At last, the switching control laws for the collision avoidance maneuver can be given. For convenience, let us take the time intervals in the i -th thrust arc for example:

$$S_i = \left\{ T_i + j\Delta T, \text{sgn}(F^*(t)) \frac{\widehat{F}\Delta T}{N} \left[\frac{N \int_{T_i+j\Delta T}^{T_i+(j+1)\Delta T} |F^*(t)| dt}{\widehat{F}\Delta T} \right] \right\}. \quad (54)$$

5. Simulation Examples

5.1. Comparison with A-Star Algorithm and Ant Colony Algorithm. In order to present some of the results of the algorithms presented, simulation examples are evaluated in some nontrivial simulation scenarios. To verify the effectiveness of the algorithm, the ant colony algorithm and A-star algorithm are compared with the algorithm proposed in this paper.

First, statistical tests are used to analyze the effectiveness of A-star, that is, to test the variance of the number of obstacles. Suppose that the path planning time of A-star algorithm and the algorithms proposed in this paper is f_1 and f_2 , and the evaluation values of the three algorithms are g_1 and g_2 . The comprehensive evaluation function can be obtained as follows:

$$E = \frac{f_1}{f_2} + \frac{g_1}{g_2}. \quad (55)$$

According to the comprehensive evaluation function, the following statistical table can be obtained as shown in Table 1.

The variance test shows that the number of obstacles has a significant impact on path planning. It can be seen from the experimental data that with the increase of obstacles, the comprehensive evaluation function tends to increase. This is because the algorithms proposed in this paper do not search all nodes, so the more the obstacles are, the more it shows its superiority.

It can be seen from Figure 2 that there are 200 obstacle nodes and black points are obstacle points and boundary points, red points are close nodes, green points are open nodes, and green line is the planned path.

Then the ant colony algorithm is compared to the algorithms proposed in this paper. Suppose that the starting position of the on-water automatic rescue intelligent robot is $[0, 20, 0]$ and the target position is $[19, 2, 0]$ as shown in Figure 3.

It can be seen from Figure 3 that when the 3-D ant colony algorithm is used for fast path planning, with

TABLE 1: Comprehensive evaluation value.

Number of obstacle nodes	50	100	150	200	250	300
Comprehensive evaluation value	0.51	0.77	1.34	1.97	2.21	2.77

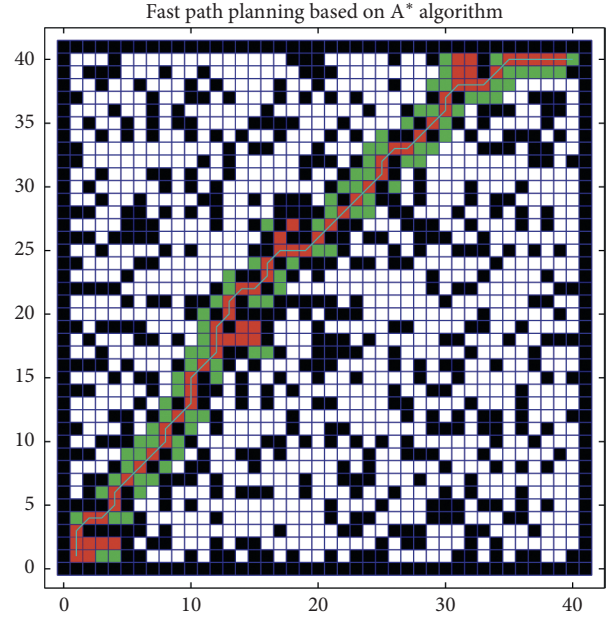


FIGURE 2: Planned path based on the A-star algorithm.

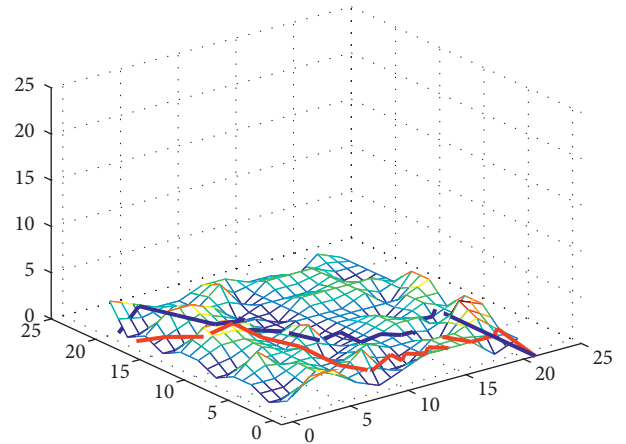


FIGURE 3: Planned path based on the ant colony algorithm.

different iteration times, the number of ants, etc., the planned path is also different. Moreover, 3-D ant colony algorithm generally only seeks the shortest path, so the planned path is not very smooth.

It can be seen from Figures 4 and 5 that when the number of iterations is the same, the best individual fitness curve varies with the number of ants. In other words, ant colony algorithm relies on prior knowledge to a large extent, so it has its shortcomings in fast path planning. To solve this problem, the mathematical model of obstacles is defined using the method proposed in this paper. Suppose that the

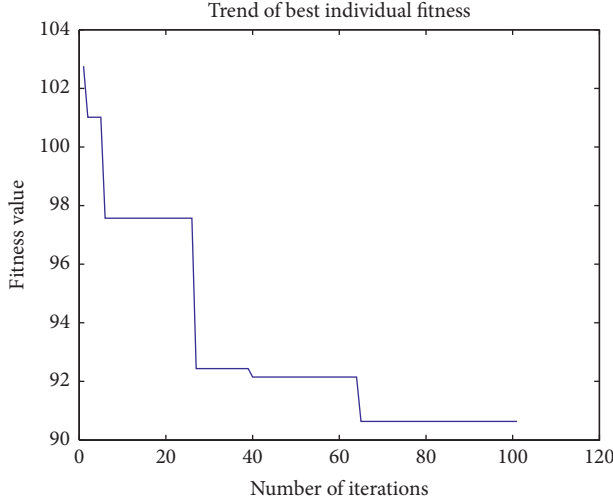


FIGURE 4: The change of best individual fitness.

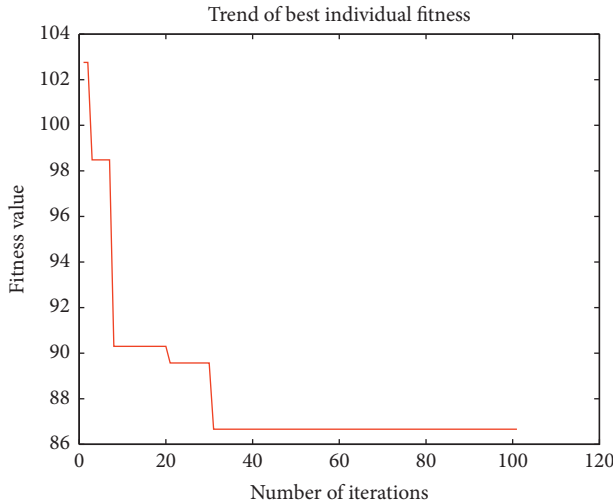


FIGURE 5: The change of best individual fitness.

starting position of the on-water automatic rescue intelligent robot is $[0, 0, 0]$ and the target position is $[40, 40, 0]$. Assume that $\xi = (x, y, z)$ is the robot position and obstacle center is $\xi_0 = (x_0, y_0, z_0)$, with axis lengths a, b, c and index parameters d, e, f . Then, construct the function as follows [16]:

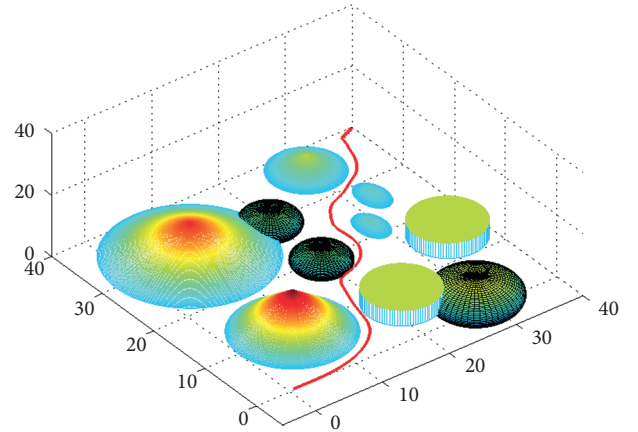
$$F(\xi) = \left(\frac{x - x_0}{a}\right)^{2d} + \left(\frac{y - y_0}{b}\right)^{2e} + \left(\frac{z - z_0}{c}\right)^{2f}, \quad (56)$$

where parameters a, b, c, d, e, f determine the shape and size of the obstacle: if $a = b = c$ and $d = e = f = 1$, the obstacle is a sphere; if $a = b$ and $d = e = 1$, $0 < f < 1$, the obstacle is regarded as a cone; if a and b are variables meeting the condition $a = b = R_1 + ((R_2 - R_1)z/c)$ and $d = e = 1$, $f > 1$ holds, the obstacle is a circular truncated cone approximately, where R_1, R_2 are radii of two bases. The obstacles are shown in Table 2, and $\rho_0(\xi) = 1$ and $\sigma_0(\xi) = 1$.

It can be seen from Figure 6 that the planned path by the improved artificial fluid method proposed in this paper can bypass the obstacle, escape from the trap area, and reach the destination rescue point smoothly.

TABLE 2: The shape parameters of each obstacle.

Obstacle	Shape	a	b	c	d	e	f	Center of obstacle
1	Sphere	3.9	3.9	3.9	1	1	1	(20, 20, 0)
2	Sphere	6	6	6	1	1	1	(30, 3, 0)
3	Sphere	4	4	4	1	1	1	(20, 30, 0)
4	Cylinder	5	5	5	1	1	10	(22, 7, 0)
5	Cylinder	5	5	5	1	1	10	(35, 15, 0)
6	Cone	2	3	2	1	1	0.3	(33, 27, 0)
7	Cone	11	11	11	1	1	0.3	(8, 30, 0)
8	Cone	8	8	13.6	1	1	0.3	(8, 10, 0)
9	Cone	5	5	5	1	1	0.3	(30, 36, 0)
10	Cone	2	3	2	1	1	0.3	(29, 22, 0)

FIGURE 6: The planned path with $(0, 0, 0)$ as starting point.

5.2. *On-Water Automatic Rescue Intelligent Robot Can Escape the Trap Area.* During the rescue process, the robot will encounter obstacles on water. It is necessary to consider the distribution of obstacles in path planning; otherwise, the on-water automatic rescue intelligent robot will fall into the dead zone of obstacle traps, making rescue not timely and effective.

It can be seen from Figure 7 that there are two planned paths for the robot. The blue curve is the shortest planned path, and the planned path does not consider how to bypass the trap area, but chose to climb directly over the obstacle; in a realistic environment, however, that would not work. The red curve is the planned path by the improved artificial fluid method proposed in this paper; it can be clearly seen that the robot can bypass the obstacle, escape from the trap area, and reach the destination rescue point smoothly.

5.3. *Comparison of Different Saturation λ and Different Starting Points.* In the process of path planning, different paths will be planned with different saturation λ . Two different planned paths are calculated based on two different λ , as shown in Figure 8.

It can be seen from Figure 8 that there are two planned paths according to different saturation λ : the blue curve is planned path 1 and the red curve is planned path 2. The saturation λ in path 1 is $1/3$ and the saturation λ in path 2 is $1/9$, and the two paths can bypass obstacles to reach the

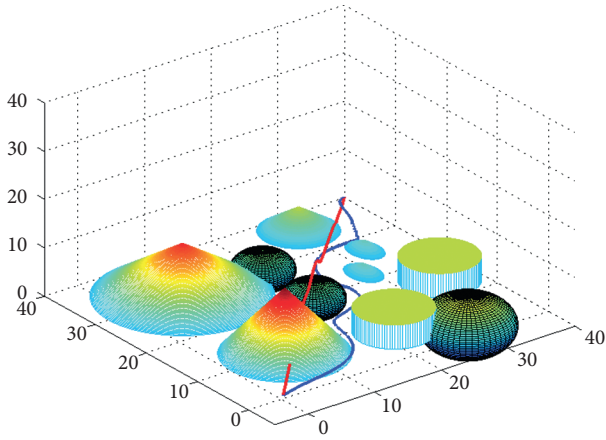


FIGURE 7: Robot bypassing the obstacle and escaping the trap area.

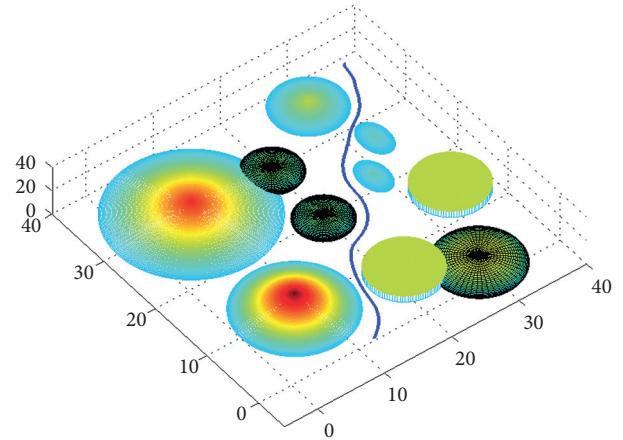


FIGURE 9: The planned path with (12, 0, 0) as starting point.

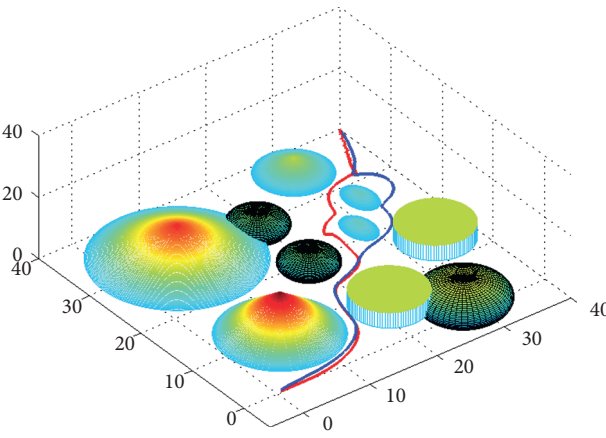


FIGURE 8: The planned path according to different saturation λ .

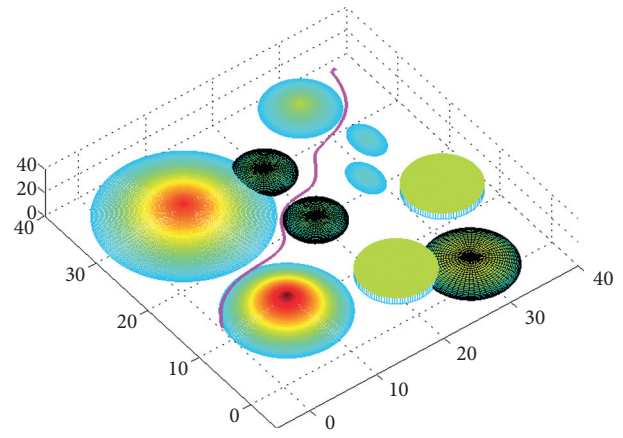


FIGURE 10: The planned path with (0, 10, 0) as starting point.

destination. It can be found that each path facing obstacles is different in the climbing height and deflection angle. Next, consider planning paths for the robot from different starting points as shown in Figures 9–11, and the destination of these planned paths is (40, 40, 0).

As can be seen from Figures 9–11, according to the planning algorithm proposed in this paper, no matter where the robot starts from, it can reach its destination smoothly. It shows the effectiveness and feasibility of the algorithm proposed in this paper.

The results in Figure 12 show the constant thrust fitting, where the red curve is the change of the ideal thrust, and the blue bars represent the size and length of the acceleration time intervals, deceleration time intervals, and zero-thrust time intervals. According to the proposed criterion in this paper, the robot should carry out impulse compensation and the size of the constant thrust impulse compensation is the same, but the time of the constant thrust impulse compensation is different. The switching control laws can be given according to the sizes and the directions of the thrust accelerations of the robot.

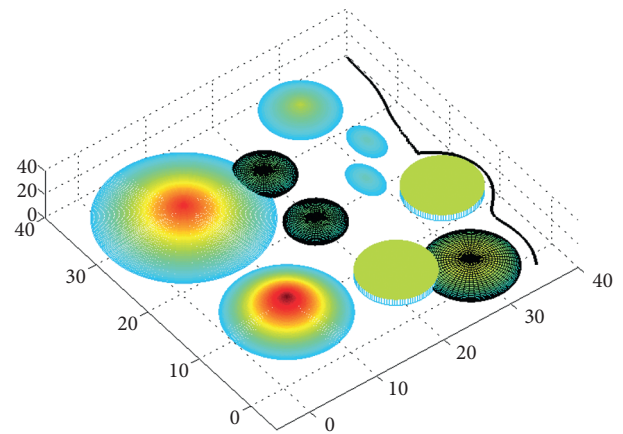


FIGURE 11: The planned path with (36, -2, 0) as starting point.

$$S = \{[0, 0, 0]; [40, 40, 0]; (3\Delta T, 2N); \dots; (36\Delta T, 3N); \dots; (58\Delta T, -2N); \dots; (60\Delta T, -7N)\}. \quad (57)$$

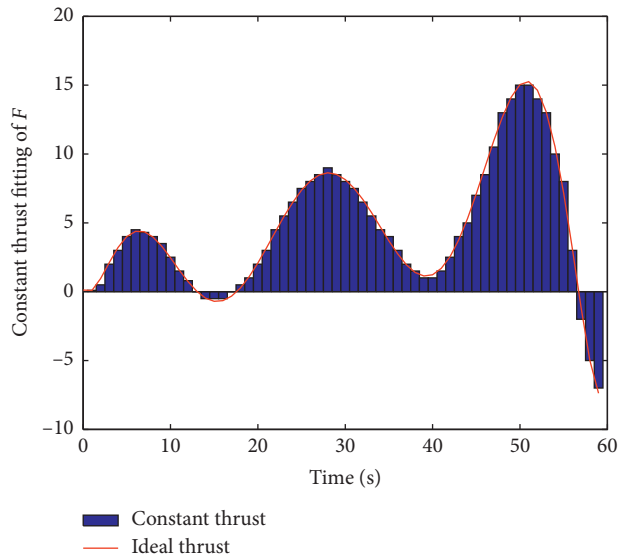


FIGURE 12: The constant thrust fitting.

6. Conclusions

This paper studies the fast path planning of on-water automatic rescue intelligent robot based on the constant thrust artificial fluid method. The tangential vector and disturbance matrix of the artificial fluid method are improved. A new constant thrust fitting algorithm is proposed through pulse compensation, and the switching control law is given. Furthermore, different parameters and different starting points are discussed separately and the simulation proves that the planned path can smoothly and quickly avoid all obstacles and finally reach the destination. In the future research work, the authors will continue to study the problem of fast path planning when there are dynamic obstacles and multiple robots working at the same time.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (no. 61304088), the Fundamental Research Funds for the Central Universities (no. 2013QNA37), the China Postdoctoral Science Foundation (no. 2015M581886), and the Jiangsu Province Postdoctoral Research Foundation (no. 1501080B).

References

[1] M. Bejiga, A. Zeggada, A. Nouffidj, and F. Melgani, "A convolutional neural network approach for assisting

avalanche search and rescue operations with UAV imagery," *Remote Sensing*, vol. 9, no. 2, p. 100, 2017.

- [2] D. Erdos, A. Erdos, and S. E. Watkins, "An experimental UAV system for search and rescue challenge," *IEEE Aerospace and Electronic Systems Magazine*, vol. 28, no. 5, pp. 32–37, 2013.
- [3] P. Rudol and P. Doherty, "Human body detection and geolocalization for uav search and rescue missions using color and thermal imagery," in *Proceedings of the 2008 IEEE Aerospace Conference*, Big Sky, MT, USA, March 2008.
- [4] T. I. Fossen and A. M. Lekkas, "Direct and indirect adaptive integral line-of-sight path-following controllers for marine craft exposed to ocean currents," *International Journal of Adaptive Control and Signal Processing*, vol. 31, no. 4, pp. 445–463, 2017.
- [5] T. I. Fossen and K. Y. Pettersen, "On uniform semiglobal exponential stability (USGES) of proportional line-of-sight guidance laws," *Automatica*, vol. 50, no. 11, pp. 2912–2917, 2014.
- [6] T. I. Fossen, K. Y. Pettersen, and R. Galeazzi, "Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 820–827, 2015.
- [7] J. Sun, B. Li, Y. Jiang, and C.-Y. Wen, "A camera-based target detection and positioning UAV system for search and rescue (SAR) purposes," *Sensors*, vol. 16, no. 11, p. 1778, 2016.
- [8] M. A. K. Jaradat, M. H. Garibeh, and E. A. Feilat, "Autonomous mobile robot dynamic motion planning using hybrid fuzzy potential field," *Soft Computing*, vol. 16, no. 1, pp. 153–164, 2011.
- [9] S. Saravanakumar and T. Asokan, "Multipoint potential field method for path planning of autonomous underwater vehicles in 3D space," *Intelligent Service Robotics*, vol. 6, no. 4, pp. 211–224, 2013.
- [10] Y. Fu, M. Ding, C. Zhou, and H. Hu, "Route planning for unmanned aerial vehicle (UAV) on the sea using hybrid differential evolution and quantum-behaved particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 6, pp. 1451–1465, 2013.
- [11] V. Roberge, M. Tarbouchi, and G. Labonte, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 132–141, 2013.
- [12] M. Chen, Q. X. Wu, and C. S. Jiang, "A modified ant optimization algorithm for path planning of UCAV," *Applied Soft Computing*, vol. 8, no. 4, pp. 1712–1718, 2008.
- [13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [14] K. Charalampous, I. Kostavelis, and A. Gasteratos, "Thorough robot navigation based on SVM local planning," *Robotics and Autonomous Systems*, vol. 70, pp. 166–180, 2017.
- [15] J. Ni, X. Li, and X. Fan, "A dynamic risk level based bio-inspired neural network approach for robot path planning," in *Proceedings of the 2014 World Automation Congress*, vol. 6, pp. 829–833, Waikoloa, HI, USA, August 2014.
- [16] P. Yao, H. Wang, and Z. Su, "UAV feasible path planning based on disturbed fluid and trajectory propagation," *Chinese Journal of Aeronautics*, vol. 28, no. 4, pp. 1163–1177, 2015.
- [17] M. Mattei and L. Blasi, "Smooth flight trajectory planning in the presence of no-fly zones and obstacles," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 2, pp. 454–462, 2010.

- [18] B. Pavol, I. željko, L. Alexander, L. Vadim, and T. Vladimir, "Solutions to the characteristic equation for industrial robot's elliptic trajectories," *Tehnickivjesnik-Technical Gzette*, vol. 23, no. 4, pp. 1017–1023, 2016.
- [19] P. Božek, M. Aiman, Al Akkad, P. Blistan, and N. Ibrahim, "Navigation control and stability investigation of a mobile robot based on a hexacopter equipped with an integrated manipulator," *International Journal of Advanced Robotic Systems*, vol. 14, no. 6, Article ID 172988141773810, 2017.
- [20] R. Pirník, M. Hruboš, D. Nemeč, T. Mravec, and P. Božek, "Integration of inertial sensor data into control of the mobile platform," in *Proceedings of the 2015 Federated Conference on Software Development and Object Technologies*, pp. 271–282, Žilina, Slovakia, November 2015.
- [21] A. Kilin, P. Bozek, Y. Karavaev, A. Klekovkin, and V. Shestakov, "Experimental investigations of a highly maneuverable mobile omniwheel robot," *International Journal of Advanced Robotic Systems*, vol. 14, no. 6, Article ID 172988141774457, 2017.