Int j simul model 15 (2016) 3, 460-472
ISSN 1726-4529 Original scientific paper

SIMULATION ON LOCAL OBSTACLE AVOIDANCE ALGORITHM FOR UNMANNED SURFACE VEHICLE

Wang, C.[*,**,#]; Mao, Y. S.[*]; Du, K. J.[*]; Hu, B. Q.[*] & Song, L. F.[*]
[*] Key Laboratory of High Performance Ship Technology (Wuhan University of Technology),
Ministry of Education, Wuhan 430063, P. R. China
[**] Research Institute of Marine System Engineering, Seoul National University, Seoul 151744, Korea
E-Mail: chriswang@whut.edu.cn ([#]Corresponding author)

Abstract
Unmanned surface vehicle (USV) is an important autonomous marine vehicle. The safe navigation of USV is directly determined by the local obstacle avoidance because it must avoid real-time local obstacle in the global path planning in a three-dimensional environment. Therefore, efficient algorithms of real-time local obstacle avoidance for USV are a critical issue. In this study, a new three-layered architecture for local obstacle avoidance algorithm was proposed to solve the local obstacle avoidance problem. First, real environment and obstacle models were established in the polar coordinate. The known static path-planning method was conducted based on particle swarm optimization (PSO). Second, the method was integrated with marine rules based on PSO. Third, an obstacle avoidance method under unknown environment was created based on rolling windows. Finally, a simulation experimental platform was developed to verify the feasibility and effectiveness of the aforementioned measure. Result shows that the proposed algorithm can effectively avoid local obstacles of USV at a computation time of less than 2 s. The USV avoids the obstacles smoothly and reaches the desired destination with complex requirements. The simulation results also demonstrate the promising application of the proposed method in studying the path planning of USV. This method can address the issues of real-time local obstacle avoidance of USV.

(Received in September 2015, accepted in March 2016. This paper was with the authors 2 months for 1 revision.)

Key Words: Unmanned Surface Vehicle (USV), Path Planning, Dynamic Collision Avoidance, Particle Swarm Optimization (PSO), Marine Rules, Rolling Windows

1. INTRODUCTION

As an autonomous ocean vehicle, unmanned surface vehicle (USV) is the only one that can play a role in water, surface, and underwater unmanned vehicle platforms at the same time [1]. The advantages of USV have been concerned gradually, and the development strategy and technology of foreign countries blockade to our country about USV. The expectation of a significant improvement in the intelligent level of USV and its wide application in military and civilian fields is rising day by day. The capability of path planning and collision avoidance of USV in a complex environment is an important embodiment of intelligence, and USV is one of the key technologies of intelligent navigation control. Its independent capability allows it to interact with the environment, which also enables it to carry out global path planning and local obstacle avoidance. In the course of navigation, USV is in an unpredictable and highly dynamic marine environment. Obstacles may appear in the global path planning, and the obstacles and their motion information can be detected and predicted by hardware (marine radar and automatic identification system). Therefore, based on global path planning, real-time local dangerous obstacle avoidance is also needed to meet the needs of USV to carry out the task smoothly. An efficient method is necessary to solve the local obstacle avoidance problem of USV.

DOI:10.2507/IJSIMM15(3)6.347

2. STATE OF THE ART

Research of local obstacle avoidance algorithm of USV started earlier in some foreign countries. Fedorenko and Gurenko [2] proposed an approach for unmanned surface vehicle motion planning in an environment with obstacles. Global planner was designed using generalized Voronoi diagrams. Local planner was implemented using the unstable regimes of control to bypass obstacles near USV. The results of simulation showed the efficiency of the proposed approaches. Liu and Bucknall [3] presented a novel computer-based algorithm to solve the problem of USV formation path planning. The algorithm was based on fast marching (FM) method and specifically designed for operation in dynamic environments through a novel constrained FM method. The algorithm was evaluated using several tests on a simulated area and was proven to work effectively in a complex navigation environment. Petres et al. [4] in France proposed a new reactive path-planning method, which considered the so called no-go zones and turned them into virtual obstacles. A potential field algorithm was proposed to realize the automatic real-time obstacle avoidance of sailing. Tang et al. [5] divided the local static obstacle avoidance algorithms into two types, namely, path-searching-based local path planning and behaviour-based reactive obstacle avoidance method. Yang et al. [6] proposed an improved A*-based algorithm, which is called direction priority sequential selection, to solve the problem of local static obstacle avoidance. The algorithm was built on a grid-based world model of the environment. However, the processes of grid-based map construction and path searching require considerable amount of time. The grid-based world model should be updated as the sensor data change. Xie et al. [7] proposed a method of the obstacle avoidance planning of USV based on improved artificial potential field; this method overcomes the problems of local minimum, destination unreachable, and poor accuracy of algorithm by using the traditional artificial potential field. Larson et al. [8], Bandyophadyay et al. [9], and Krishnamurthy et al. [10] used behaviour-based reactive obstacle avoidance approaches to deal with the local obstacle avoidance for USV. Tang et al. [11] developed a behaviour-based reactive local obstacle avoidance algorithm based on a dynamic window. Zhuang et al. [12] designed a planning method based on the marine radar image processing and used the Dijkstra algorithm to search for partial optimal paths.

Although previous studies achieved advances in local dangerous obstacle avoidance of USV, some shortcomings remain to be addressed. For instance, the obstacle was assumed circular in previous studies. This assumption is convenient to solve the problem, but it is unreasonable because some obstacle avoidance paths would be lost. Moreover, the algorithms are easy to be trapped in local optimum by assuming the obstacle as circular.

Based on the analysis of various USV obstacle avoidance methods, a three-layered architecture for local obstacle avoidance algorithm is proposed in this study. The known static path-planning method based on particle swarm optimization (PSO), the known dynamic path-planning method integrated with the marine rules based on PSO, and the obstacle avoidance method under unknown environment based on rolling windows are considered.

The remainder of this paper is organized as follows. Section 3 describes the methodology about the three-layered architecture for the local obstacle avoidance algorithm. Section 4 presents a simulation experiment to evaluate the algorithm performance. Section 5 shows the vision simulation platform. Section 6 concludes this study.

3. METHODOLOGY

3.1 Known static path planning based on PSO

During the global path planning in real sea, USV can detect local obstacles that cannot be marked on an electronic chart by hardware (such as automatic identification system and nautical radar). Thus, static obstacles that are not considered in the global path planning should be avoided, and the destination point or sub target in this situation should be rapidly and safely reached.

Modelling under polar coordinates

The polar coordinate system can associate PSO in each dimension component and path at the desired speed and heading. This system is also sensitive to the length and angle and facilitates the obstacle information. Therefore, the polar coordinate system is utilized to build the model. The starting and end points of the polar coordinate system are the starting and end points of the selected sub target segment in the global planning path, respectively. The line that connects the two points is the polar axis of the polar coordinate system, and the starting point is the pole. The counter clockwise direction is positive.

In a real environment, the obstacle shapes vary. Shapes scanned by radar, sonar, or senor are complex. A high cost will be incurred because the calculation process requires considerable computation time if directly using the untreated obstacle shapes. To improve the computation performance, before the path planning, giving a reasonable simplification of the obstacles and dealing with the bounding box of the obstacles are necessary.

Most of the domestic scholars have only taken the surrounding to deal with the obstacle bounding volume. Ships on sail are not considered obstacles, which length-to-width ratio is large. In this case, only taking the surrounding will lose some areas for navigation and lead to an unreasonable strategy of obstacle avoidance. In this study, the obstacle bounding volume processing includes two ways as follows: (1) for a small length-to-width ratio (i.e., less than 2:1), using circular obstacle does not lose many feasible paths, and it greatly simplifies the calculation and can effectively shorten the calculation time; (2) for a large length-to-width ratio (greater than 2:1), more realistic obstacle shapes are considered in the simulation (i.e., a bounding box and oriented bounding box (OBB) [9]).

Many uncertainties occur during USV sailing in the sea, such as the uncertainties in the movement of obstacles, the motion of the unmanned boat, the execution of the agency, and unmanned submarine positioning. The obstacle detection system also exhibits inevitable errors. Considering such uncertainty factors and to simplify the calculation, this study regards USV as a centre of mass. Accordingly, the model of obstacles is expanded.

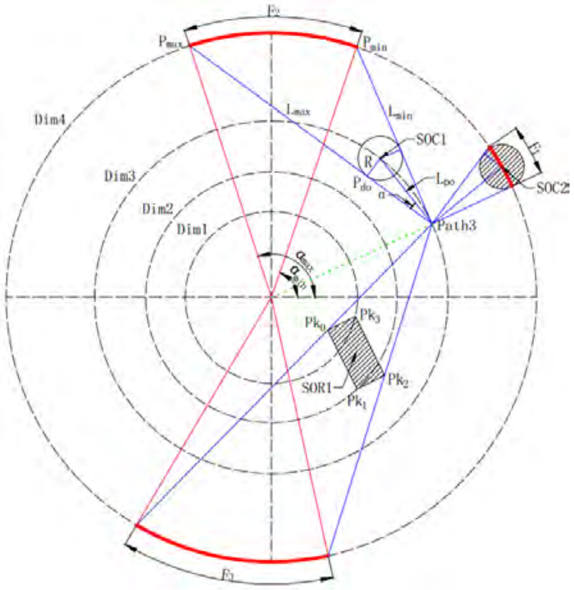


Figure 1: Schematic of solving obstacle forbidden angle.
(*Dim*: Dimension circle; Path: path point; *SOC*1: known static circular obstacle; *SOR*: known static rectangular obstacle; *F*: forbidding domain; $\alpha_{\max}$: forbidden angle)

Path feasibility judgment

In the process of path planning, USV is not allowed to enter one part of the arc segment because the obstacle is ahead of the dimension. This arc segment is called the off-limit domain of the dimension. As shown in Fig. 1, we assume that USV is moving to Path 3. The arc F_1 and F_2 on *Dim*4 are the off-limit domains generated by *SOC*1 and *SOC*2, respectively. F_3 is the off-limit domain generated by *SOR*1. The dynamic calculation method to deal with the forbidden domain is adopted to make the calculation efficient. When calculating the path of one point, only the off-limit domain generated by the obstacles before the dimension needs to calculate, without considering the obstacles behind.

To facilitate the calculation of the polar angle of the path points, the forbidding domain needs to be transformed into the corresponding forbidden angle of the polar coordinates. Forbidden angle is the angle between polar axis and the line that connects pole and forbidding domain. The forbidden angle of static circular and rectangular obstacles is calculated as follows:

(1) Solving the forbidden angle of the known static circular obstacle

The forbidden angle of a circular obstacle can be divided into two parts as follows: 1.The forbidden angle of obstacle before the dimension is solved; 2. if the obstacle is the last one, it also needs to calculate the final termination point between one and the forbidden angle. The second part stands for the angle along with the confirmed environmental information and the solution in the cycle of path optimization algorithm, thereby minimizing the required computation. The first part is changed by the path optimization algorithm and needs to be done within the loop algorithm.

As shown in Fig. 1, under the polar coordinate system, *SOC*1 is in the triangle composed of path point Path3, obstacle circle point O_1, and tangent point P_{do} through Path3 tangent to obstacle circle. The length of line L_{po} is obtained according to Path3 and O_1. According to the angle α of L_{po} and tangent, $\alpha = \arcsin\frac{R}{L_{po}}$, the polar angle of $L_{\min}$ and $L_{\min}$ can be obtained. Converting into the Descartes coordinate system can yield the tangent equation. Through the location of the tangent point P_{do}, we can acquire the coordinate value of the end points $P_{\min}$ and $P_{\max}$ of the forbidden domain, as well as the banned angles $\alpha_{\min}$ and $\alpha_{\max}$.

(2) Solving the forbidden angle of the known static rectangular obstacle

Rectangular barrier is also divided into two parts. The processing method is same as that for circular obstacles. The rectangular obstacle needs to confirm to which region of the nine rectangular obstacles the path point belongs. The region-partitioning coding of directed bounding box is shown in Fig. 2. The point *SOC* in Fig. 1 locates in the rectangle of code 3. The two endpoints of off-limit domain are the intersections of dimension circle with the connection of Path3 and point 0 and the connection of Path3 and point 2.

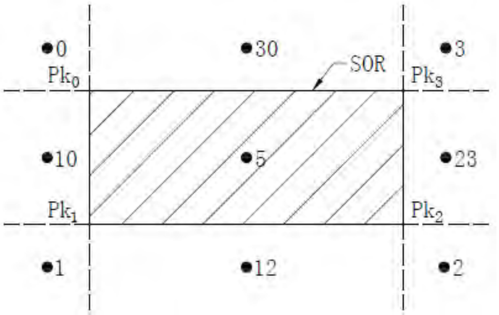


Figure 2: Schematic of the region-partitioning coding of directed bounding box.

For rectangular obstacles, two selections of dimension point exist. If we take the minimum and maximum diameter points of four comer points, the forbidden domain will be significantly large when we calculate the path point in the next dimension, thus resulting in an unreasonable path. If we take two points in the middle of the pole diameter, the amount of calculation is increased, but the optimal path is reasonable.

Determination of the PSO algorithm and fitness function

PSO algorithm is a cluster optimization algorithm proposed by Eberhart and Kennedy in 1995. Given a group of random particles at a certain speed, the current optimal solutions of each particle found by itself are $p_{i,d}$ and the global optimal solutions are $p_{g,d}$ where $d = 1, 2, ..., N$, and N is the particle swarm iteration number [13-15]. After finding two optimal solutions, the velocity and position of the particles are respectively updated by Eqs. (1) and (2).

$$v_{i,d}(t+1) = \omega v_{i,d}(t) + c_1 r_1 \left(p_{i,d} - x_{i,d}(t)\right) + c_2 r_2 \left(p_{g,d} - x_{i,d}(t)\right) \tag{1}$$

$$x_{i,d}(t+1) = x_{i,d}(t) + v_{i,d}(t+1) \tag{2}$$

Type (1): r_1 and r_2 are random numbers of the uniform distribution of 0 to 1. ω is the inertia weight; in this study, its value selects a linear decreasing inertia weight that is widely used [11]. As the value of learning factor, c_1 and c_2 are valued 2.0 by many experiments in this study.

The requirement of performance index in this study is the shortest feasible path without intersection with obstacle. Therefore, the path length value is used as the fitness function value, as shown in Eq. (3), which indicates the distance between points $i+1$ and i. The fitness function value of the feasible path is the length of the optimal path, whereas the fitness function value of the infeasible path is set to infinity.

$$Fit(i) = \begin{cases} L_{SP_1} + \sum_{i=1}^{n-1} L_i L_{i+1} + L_{P_n G} & L_{P_i P_{i+1}} \cap NPD = \varnothing \\ +\infty & L_{P_i P_{i+1}} \cap NPD \neq \varnothing \end{cases} \tag{3}$$

Path-planning algorithm based on PSO

In a complex environment, the path-planning algorithm of USV is easy to be trapped in the local optimum because many obstacles can lead to the large dimension of PSO. Strong mobility of USV is needed if many inflection points exist on the path. Therefore, the method of path-smoothing optimization is cited. From the starting point link, in which every path points back to the starting point reversely, if the wired segment does not intersect with obstacles, the wired segment is selected as a sub path segment. The value of the middle path point of sub path segment is then calculated, and the optimal path array is provided until to the penultimate path point. Sometimes, the path is necessary to optimize many times until the fitness function value is no longer changed.

3.2 Path planning in known dynamic environment

Known dynamic obstacle avoidance model

The geometry models of USV and circular dynamic obstacles are shown in Fig. 3. V_{USV} is the velocity of USV, and V_{Obs} is the velocity of obstacle. α is the angle of the polar axis to VUSV, and it can be expressed as $\alpha = \angle(V_{USV}, e_x)$, where e_x represents the polar axis. Similarly, $\beta = \angle(V_{Obs}, e_x)$ is the angle from the polar axis to V_{Obs}. $\theta = \angle(L_{RO}, e_x)$is the angle from polar axis to LRO, which is the line from USV to the centre of the circle obstacle. $\varphi = \angle(V_{USV}, \Delta V)$ is the angle from ΔV to VUSV. $\gamma = \angle(\Delta V, L_{RO})$, $\mu = \angle(L_{RO}, tanL)$, and $\tan L$ represents the circle obstacle tangents.

For the USV avoiding obstacle interval in the next time, γ should be out of the $(L_{RO}-\mu, L_{RO}+\mu)$. The conditions of the speed and steering can be found by solving γ, which achieves the USV avoidance.

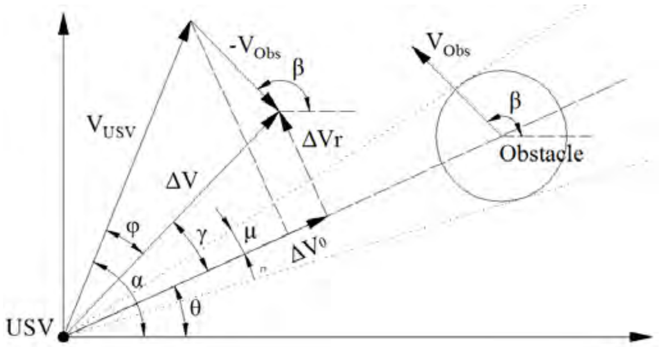


Figure 3: Obstacle avoidance model and decomposition of relative speed ΔV.

In Fig. 3, ΔV is divided into two directions; one is ΔV_0, which refers to the centre of the circle obstruction, and the other is ΔV_γ, which is vertical to ΔV_0. ΔV_0 makes USV move to obstruction, whereas ΔV_γ makes USV move away from obstruction.

The speed of the triangle is made up by V_{USV}, V_{Obs}, and ΔV, and it can solve ΔV_r and γ.

$$\Delta\gamma = \frac{-\sin\varphi}{\Delta v}\Delta v_{USV} + \frac{v_{USV}\cos\varphi}{\Delta v}\Delta\alpha \tag{4}$$

To make USV flee the danger zone between the two tangents, $\Delta\gamma$ needs to satisfy the following inequality Eq. (5):

$$\begin{cases} \Delta\gamma \geq \mu - r, & \gamma > 0 \\ \Delta\gamma \leq -(\mu + r), & \gamma < 0 \end{cases} \tag{5}$$

Eq. (4) shows that $\Delta\gamma$ is a function about speed change amount ΔV_{USV} and heading change amount $\Delta\alpha$. Therefore, the task of avoidance transforms into adjusting these two values. The establishing and solving processes of the rectangle obstacle model are similar to those for circular obstacle.

Constraint of the International Regulations for Preventing Collisions at Sea

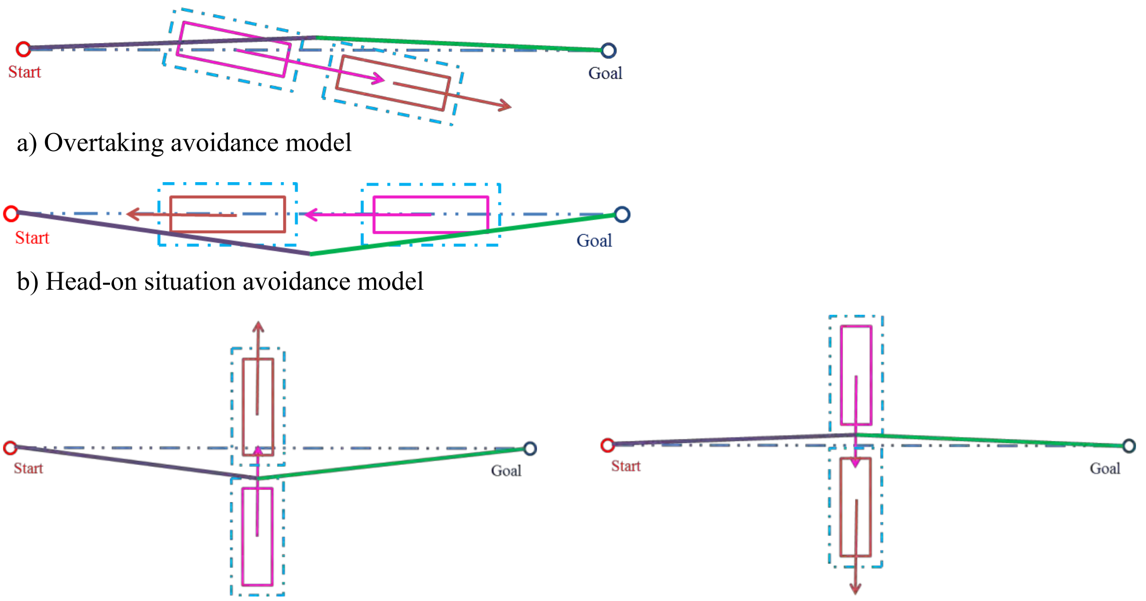
The International Regulations for Preventing Collisions At Sea (COLREGS) [16] was developed by the International Maritime Organization to prevent and avoid marine ship collision. Making USV comply with COLREGS is a reasonable solution given no specific laws and regulations on USV sailing.

The Rules 13, 14, and 15 in the Convention of COLREGS [16] make the provision for the possible collision situation on overtaking, head-on situation, and crossing situation, respectively. The following shows the USV actual situation according to related regulations in COLREGS. Fig. 4 presents the avoidance model, which shows four situations. Obstacle A is in the position when USV is at the starting point "Start," and the arrow points in the direction of its movement. Obstacle B is in the position when USV completes avoidance.

Known dynamic path-planning algorithm based on PSO

a) Determining the fitness function based on PSO

In most USV avoidance algorithm research, USV speed does not change but only heading changes. However, in some situations, speed and heading need to change simultaneously to achieve avoidance. This study uses the strategy of changing speed and course simultaneously. The USV path-planning problem can then be expressed as a target optimization problem under multiple conditions, as shown in Eq. (6).



a) Overtaking avoidance model

b) Head-on situation avoidance model

c) Starboard crossing avoidance

d) Port crossing avoidance

Figure 4: Avoidance model.

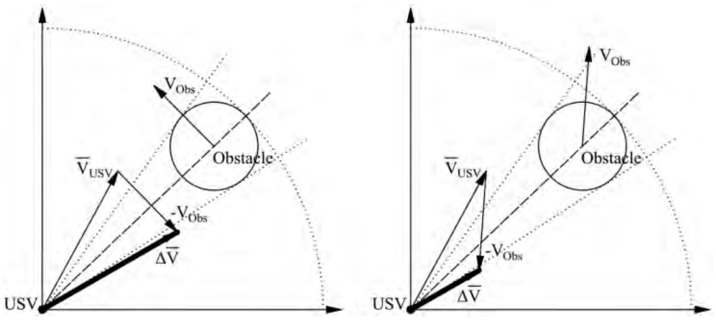
$$\begin{cases} f(\Delta v_{USV},\Delta\alpha)=m_1|\Delta v_{USV}|+m_2|\Delta\alpha| \\ \frac{-\sin\varphi}{\Delta v}\Delta v_{USV}+\frac{v_{USV}\cos\varphi}{\Delta v}\Delta\alpha >= \mu-\gamma \qquad (\gamma>=0) \\ \frac{-\sin\varphi}{\Delta v}\Delta v_{USV}+\frac{v_{USV}\cos\varphi}{\Delta v}\Delta\alpha < \mu-\gamma \qquad (\gamma<0) \end{cases} \tag{6}$$

where m_1 and m_2 represents the weights of Δv_{USV} and $\Delta\alpha$, respectively; $f(\Delta v_{USV},\Delta\alpha)$ is fitness function. Finally, the optimal solution should be in the feasible region, and it should satisfy COLREGS. If multiple known dynamic obstacles exist in the path, the second and third constraints in Eq. (6) should be added.

b) Determining the fitness function based on PSO

The above algorithm achieves optimal speed and heading after obstacle avoidance, and the end point of the obstacle avoidance should be determined.

If the obstacle is to run toward the USV, its global polar radius is reducing before avoidance does not reach the end point. In Fig. 5 a, $\bar{V}_{usv}$ is the optimized USV speed, and $\Delta\bar{V}$ is the resultant velocity of the optimized USV speed and obstacle speed. When the obstacle is regarded still in the avoidance model, the time consumed to complete avoidance is t if USV reaches the remote of the obstacle as a resultant velocity $\Delta\bar{V}$. In the global coordinate system, the polar radius and the polar angle of the avoidance end point after t can be calculated.



a) Crossing situation

b) Overtaking

Figure 5: Avoidance end point.

If the obstacle is to run away from the USV, its global polar radius is increasing before avoidance does not reach the end point, as shown in Fig. 5 b. In this case, if we still process as described above, avoidance end point runs away from the target point.

3.3 Avoidance strategy in unknown dynamic environment

Avoidance strategy in unknown dynamic environment

USV scrolling window is a fan-shaped area with a radius that is the detection range R of obstacle avoidance sonar. At t, the USV scrolling window is $\mathrm{Win}(P(t))=\{P|P\in W, d(P,P(t))\leq R\}$ in $P(t)$. In this equation, R is greater than the moving step length of USV, and W is the feasible region after the first two layers of the algorithm.

If the USV speed is V and the control period is T, the moving step length in a period is $\varepsilon = T\times\Delta V$ (ΔV is the USV average speed in this period). The radius R of scrolling window needs to satisfy the condition $0<\varepsilon<R$.

The obstacle is also surrounded and expanded in this layer algorithm. At t, the centre coordinate of unknown round dynamic obstacles is $(\rho_i(t), \theta_i(t))$, and the radius is r_i. If the centre of the round obstacles moves $\Delta r = V_i\times T$ (V_i is the speed of the i^{th} obstacle detected) along the direction of the obstacle moving, the centre coordinate of the expansion model is $(\rho_i'(t), \theta_i'(t))$, and the radius is $r_i'=r_i+\dfrac{\Delta r}{2}$.

For dynamic unknown rectangular obstacle UOR_i, the two corners in front of the movement direction should move $\Delta d = V_i\times T$ into the moving and perpendicular directions in the local coordinate. The two corners behind the movement direction should move Δd into the perpendicular direction. After the local coordinate system is converted into the global coordinate system, the four-corner point coordinate can be obtained at the polar coordinate system.

The feasible region in the USV window can be expressed as:

$$WFD(t+\Delta t)=Win\left(P_R(t)\right)\cap\left(\bigcap_{i\in SOnum}\left(SO_i^o\right)^c\right)\cap\left(\bigcap_{j\in DOnum}\left(DO_j^o(t+\Delta t)\right)^c\right)\cap\left(\bigcap_{k\in DOUnum}\left(DO_k^o(t+\Delta t)\right)^c\right) \tag{7}$$

where the superscripts "o" and "c" and represent the internal and complement of each respective set, SO_{num} is the number of static obstacles, DO_{num} is the number of known dynamic obstacles, and DOU_{num} is the number of dynamic unknown obstacles.

Avoidance strategy in unknown environment based on rolling window

For the possible mutation of obstacle trajectory, the algorithm in this study analyses the obstacle movement of multi-cycle in advance. By predicting the location of USV and obstacles, we can judge the four states that USV might encounter as follows:

(1) Dangerous state: If USV moves forward at the speed and direction, it will enter the obstacle forecast expansion region in the next cycle T_0 and might then collide with an obstacle. Avoidance measures must be taken immediately in this case.

(2) Quasi dangerous state: If USV moves forward at the speed and direction, it is safe in the next cycle T_0. Meanwhile, the USV will enter the obstacle forecast expansion region in the cycle $(T+2T_0)$. Avoidance path should thus be solved using the avoidance algorithm based on scroll window according to the forecast expansion region in the $(T+2T_0)$ period.

(3) Potentially dangerous state: If USV continues to move forward at the current state, it will not enter into an obstacle corresponding forecast expansion area in the window. As a result, the USV does not need to take any avoidance strategy.

(4) Safe state: If no any dynamic unknown obstacles exist in the USV scrollable window or dynamic obstacle is predicted to move away from USV in the scrollable window, then the USV will not collide at the current motion state and is therefore safe.

4. RESULTS ANALYSIS AND DISCUSSION

4.1 Analysis of simulation results

Assuming that USV sails to a certain area, the starting point of local dangerous obstacle avoidance algorithm "Start" and "Goal" is designated by the global path optimization algorithm in this section. At this point, USV goes into the local dangerous obstacle avoidance mode. The distance between subpath segments "Start" and "Goal" is 3600 m, and the speed is 40 kn. In the PSO algorithm, the parameters $\omega_{\max} = 0.95$, $\omega_{\min} = 0.20$, and $c_1 = c_2 = 2$. In the first layer of path planning when a static environment is known, the number of iterations is 100, and the particle number is 50. On the second known path planning in dynamic environment, $m_1 = 1$, and $m_2 = 70$. The particle swarm iteration time is 50, and the particle number is 50. Based on the rolling windows in the third layer of obstacle avoidance in unknown dynamic environment, the radius of the rolling window is 800 m, and the control cycle is 2 s. The path section of the local obstacle avoidance is described below.

Path planning on known static environment

In this path, three known static circular obstacles (*SOC*1, *SOC*2, *SOC*3) and two known static rectangular barriers (*SOR*1, *SOR*2) exist. The relevant parameters are shown in Tables I and II. Each particle has a total of seven dimensions. After calculating the first layer of path-planning algorithm, the optimal path is 3721.65 m, which takes 1.498 s, as shown in Fig. 6. The path planning of the node is shown in Table III.

Table I: Known static circular obstacle parameters.

Number	**Polar diameter** (m)	**Polar angle** (rad)	**Diameter** (m)
1	1000.00	0.000	400
2	2000.00	0.000	800
3	2630.59	6.131	200

Table II: Static rectangular barrier known *SOR* parameters.

Number	**Centre polar diameter** (m)	**Centre polar angle** (rad)	**Half long** (m)	**Half width** (m)	**Direction angle** (°)
1	3394.113	5.489	112	20	140
2	4081.478	5.858	140	16	90

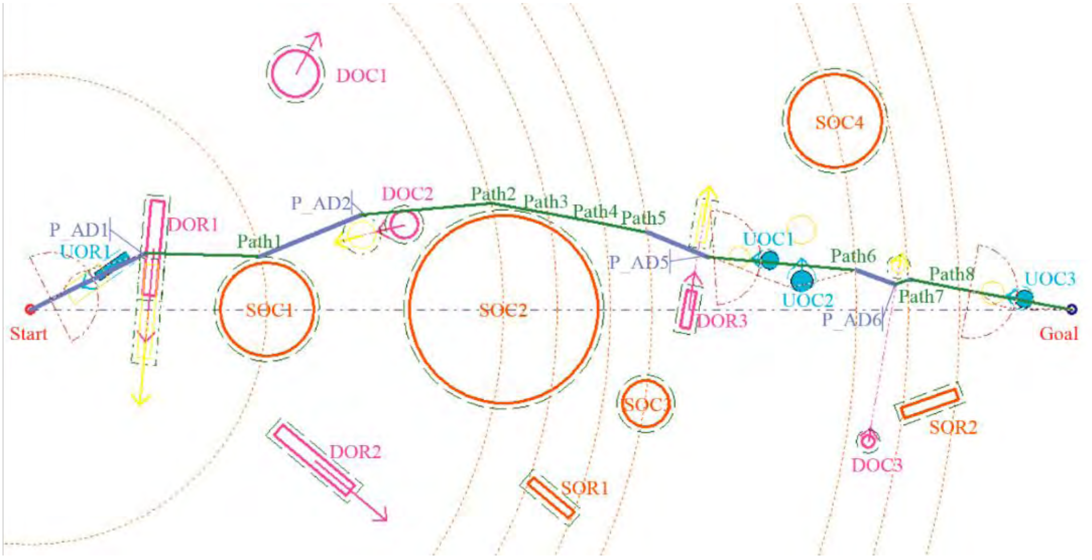


Figure 6: Known static path planning based on PSO.

Table III: Known static path-planning node.

Path to the point	**Polar diameter** (m)	**Polar angle** (rad)	**Path to the point**	**Polar diameter** (m)	**Polar angle** (rad)
1	1000.00	0.237	5	2630.59	0.107
2	2000.00	0.237	6	3465.36	0.013
3	2229.95	0.180	7	3532.08	0.007
4	2455.24	0.136	8	3930.80	0.022

Path planning on known dynamic environment

USV in the "Start" first detects the known dynamic rectangular barrier *DOR*1 (a container ship; related parameters are shown in Table IV). And USV meets *DOR*1 at 252.03°, which is the crossing situation in maritime rules. The USV then needs to get away from the rear of *DOR*1, namely, the USV should sail to left. The second known path-planning algorithm in dynamic environment takes 0.109 s. The obstacle avoidance method is to alter course to port 13.01°, and the travel speed is constant. The related parameters are shown in Table V. After running to the end points of obstacle avoidance, the USV successfully avoids obstacle *DOR*1 and then turns toward subtitle punctuation Path1.

Table IV: Known dynamic rectangular barrier *DOR*1 parameters.

Number	**Centre polar diameter** (m)	**Centre polar angle** (rad)	**Half long** (m)	**Half width** (m)	**Direction angle** (°)	**Speed** (kn)
1	1521.184	1.156	200	28	265	20

Table V: Known dynamic environment obstacle avoidance path node and quantity change.

Path point	**Polar diameter** (m)	**Polar angle** (rad)	**Speed** (kn)	**Calculating time** (s)
0	539.77	0.453	20	0.109

USV detects the known dynamic rectangular obstacle *DOR*2 in the first path at point Path1 grade (container ships), and dynamic circular obstacles *DOC*1 and *DOC*2 are known. The second algorithm judges that *DOC*1 and *DOR*2 will not crash into the USV. The meet angle between *DOC*2 and unmanned craft is 182.03°, which is the meet situation in maritime rules. The USV needs to navigate to the left from *DOC*2. Although the second algorithm takes 0.187 s, the obstacle avoidance method turns to port to 13.01°, and the travel speed is constant. The unmanned craft, after running to the end points of obstacle avoidance, successfully avoids obstacle *DOC*2 and then turns to Buddhism punctuation Path2. The unmanned craft detects known dynamic rectangular barrier *DOR*3 at Path5 and "P_AD5" dynamic path points are calculated by the second algorithm.

Obstacle avoidance strategy in unknown dynamic environment

USV continually detects the unknown obstacle since starting point "Start." When USV runs into (pole size: 539.78 m, polar angle: 0.453 rad) position to detect unknown dynamic rectangular barrier *UOR*1 related parameters (see Fig. 6 and Table VI), obstacle avoidance begins based on rolling windows. After determining that USV is in a potentially dangerous state, such as listed in Table VII, in the first cycle the polar angle becomes constant, and the USV continues to travel. In the second cycle, the USV is in a quasi-dangerous state. The obstacle avoidance module is calculated, and the course is adjusted. By the end of the third cycle, marching *UOR*1 completely disappears in the rolling window of unmanned craft, and the unmanned craft completes *UOR*1 obstacle avoidance. In the fourth cycle, the unmanned craft is along the subtitle punctuation Path1 direction.

Table VI: Unknown dynamic rectangular barrier parameters.

Number	**Centre polar diameter** (m)	**Centre polar angle** (rad)	**Half long** (m)	**Half width** (m)	**Direction angle** (°)	**Speed** (kn)
1	600	0.00	120	32	180	30

Table VII: "Start" and Path1 unknown dynamic obstacle avoidance path among nodes.

Number	**Polar angle** (rad)	**Number**	**Polar angle** (rad)	**Number**	**Polar angle** (rad)
1	0.453	3	0.981	5	0.318
2	0.981	4	0.632	——	——

Way forward movement detects *UOC*1 and *UOC*2 by using the third layer of the path-planning algorithm and dotted line, as shown in Fig. 6.

5. VISION SIMULATION PLATFORM FOR THREE LOCAL OBSTACLE AVOIDANCE SYSTEMS ABOUT USV

In this study, the basic idea about the three local obstacle avoidance systems on USV is as follows: with the known static path-planning algorithm based on the optimal path planning of the local static and basis to sail forward and by observing the external environment, the USV travels while avoiding real-time obstacles.

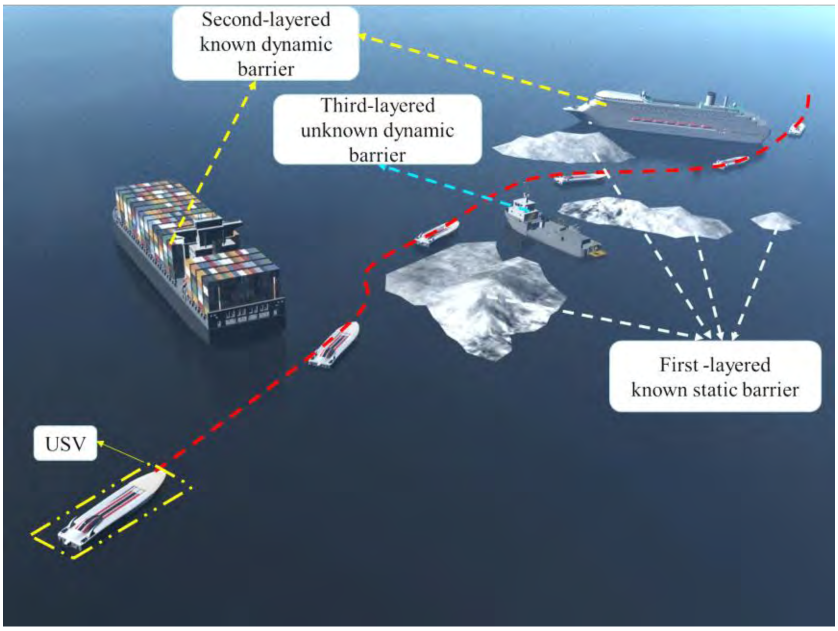

Second-layered known dynamic barrier
Third-layered unknown dynamic barrier
First -layered known static barrier
USV

Figure 7: Vision simulation platform of USV.

The vision simulation platform (Fig. 7) of USV is developed by Unity3D based on the aforementioned algorithm. Navigation demonstration is realized by collecting resource data, monitoring the status of all navigation ships and load simulation environment according to the demand area, recording the simulation motion data, and saving them to a file. The demonstration shows that the USV smoothly passes through the obstacles.

6. CONCLUSION

To resolve the problems of local obstacle avoidance for USV, this study proposes three layers of the local dangerous obstacle avoidance path optimization algorithm by using polar coordinates system to build the model and incorporating maritime rules into the known PSO-based dynamic path planning algorithm. The proposed technique is a good solution to solve the problems of local obstacle avoidance for USV under an environment with static and dynamic obstacles. The conclusions were formulated as follows.

(1) The calculation time for simulation was only 1.498 s in the first layer of known static path-planning based on PSO; hence, shortening the calculation time and reducing the turning points for generating an improved path by adopting heuristic knowledge can initialize the path of path-planning algorithm based on PSO.

(2) Based on the simulation results of the second known path-planning algorithm, the USV smoothly passes through the obstacles from the rear of the dynamic rectangular obstacle (*DOR*1) and the starboard of the circular obstacle (*DOC*2); the USV satisfies the maritime rules about the head-on situation avoidance and starboard crossing avoidance. The results indicated that the navigation for USV safety must be completed by incorporating maritime rules into the known dynamic path planning algorithm based on PSO.

(3) In the simulation, when the USV was under the unknown dynamic environment, the obstacle avoidance module was calculated and the course was adjusted quickly to sail safely through unknown dynamic rectangular (*UOR*1) in a quasi-dangerous state. The simulation experiments verified that the proposed method could effectively solve the problems of local obstacle avoidance for USV under the unknown dynamic environment by using polar coordinates model to make each dimension in PSO associated with the desired and heading in the path.

The proposed algorithm has a significant potential to solve the local dangerous obstacle avoidance problem of USV. However, some limitations exist because the proposed algorithm is only aimed at the path-planning level without considering the control system of USV. In the future, the control system of USV should be considered into the local obstacle avoidance algorithm.

REFERENCES

[1] Yan, R.-J.; Pang, S.; Sun, H.-B.; Pang, Y.-J. (2010). Development and missions of unmanned surface vehicle, *Journal of Marine Science and Application*, Vol. 9, No. 4, 451-457, doi:10.1007/s11804-010-1033-2

[2] Fedorenko, R.; Gurenko, B. (2016). Local and global motion planning for unmanned surface vehicle, *MATEC Web of Conferences – Proceedings of the 3rd International Conference on Control, Mechatronics and Automation (ICCMA 2015)*, Vol. 42, Paper 01005, 6 pages, doi:10.1051/matecconf/20164201005

[3] Liu, Y.; Bucknall, R. (2015). Path planning algorithm for unmanned surface vehicle formations in a practical maritime environment, *Ocean Engineering*, Vol. 97, 126-144, doi:10.1016/j.oceaneng.2015.01.008

[4] Petres, C.; Romero-Ramirez, M.-A.; Plumet, F. (2011). Reactive path planning for autonomous sailboat, *Proceedings of the 15th International Conference on Advanced Robotics*, 112-117, doi:10.1109/ICAR.2011.6088585

[5] Tang, P.; Zhang, R.; Liu, D.; Huang, L.; Liu, G.; Deng, T. (2015). Local reactive obstacle avoidance approach for high-speed unmanned surface vehicle, *Ocean Engineering*, Vol. 106, 128-140, doi:10.1016/j.oceaneng.2015.06.055

[6] Yang, A.; Niu, Q.; Zhao, W.; Li, K.; Irwin, G. W. (2010). An efficient algorithm for grid-based robotic path planning based on priority sorting of direction vectors, Li, K.; Fei, M.; Jia, L.; Irwin, G. W. (Eds.), *Life System Modeling and Intelligent Computing*, Springer-Verlag, Berlin, 456-466, doi:10.1007/978-3-642-15597-0_50

[7] Xie, S.; Wu, P.; Peng, Y.; Luo, J.; Qu, D.; Li, Q.; Gu, J. (2014). The obstacle avoidance planning of USV based on improved artificial potential field, *2014 IEEE International Conference on Information and Automation (ICIA)*, 746-751, doi:10.1109/ICInfA.2014.6932751

[8] Larson, J.; Bruch, M.; Ebken, J. (2006). Autonomous navigation and obstacle avoidance for unmanned surface vehicles, *SPIE Proceedings, Unmanned Systems Technology VIII,* Vol. 6230, Paper 623007, 46-52, doi:10.1117/12.663798

[9] Bandyophadyay, T.; Sarcione, L.; Hover, F. S. (2009). A simple reactive obstacle avoidance algorithm and its application in Singapore Harbor, Howard, A.; Iagnemma, K.; Kelly, A. (Eds.), *Field and Service Robotics – Results of the 7th International Conference*, Springer-Verlag, Berlin, 455-465, doi:10.1007/978-3-642-13408-1_41

[10] Krishnamurthy, P.; Khorrami, F.; Ng, T. L. (2008). Obstacle avoidance for unmanned sea surface vehicles: a hierarchical approach, *IFAC Proceedings Volumes – Proceedings of the 17th World Congress on the International Federation of Automatic Control*, Vol. 41, No. 2, 6798-6803, doi:10.3182/20080706-5-KR-1001.01152

[11] Tang, P.; Qiao, L.; Zhang, R. (2011). Near-field reactive obstacle-avoidance for USV, *Journal of Huazhong University of Science & Technology (Natural Science Edition)*, Vol. 40, No. S2, 400-402, doi:10.13245/j.hust.2011.s2.105

[12] Zhuang, J.-Y.; Su, Y.-M.; Liao, Y.-L.; Sun, H.-B. (2012). Unmanned surface vehicle local path planning based on marine radar, *Journal of Shanghai Jiaotong University (Science)*, Vol. 46, No. 9, 1371-1375

[13] Wahyudi, E. J.; Santoso, D.; Kadir, W. G. A.; Alawiyah, S. (2014). Designing a genetic algorithm for efficient calculation in time-lapse gravity inversion, *Journal of Engineering and Technological Sciences*, Vol. 46, No. 1, 58-77, doi:10.5614/j.eng.technol.sci.2014.46.1.4

[14] Wu, P.; Liu, W.; Zhang, J. (2014). Economic dispatch problem based on improved particle swarm optimization, *Journal of Engineering Science and Technology Review*, Vol. 7, No. 1, 126-131

[15] Wu, Z. (2014). Optimization of distribution route selection based on particle swarm algorithm, *International Journal of Simulation Modelling*, Vol. 13, No. 2, 230-242, doi:10.2507/IJSIMM13(2)CO9

[16] The International Maritime Organisation. (1972). Convention on the international regulations for preventing collisions at sea, 25-27 from *http://www.imo.org/Conventions/mainframe.asp?topic_id=257&doc_id=649*, accessed on 08-07-2015