

Autonomous obstacle avoidance of an unmanned surface vehicle based on cooperative manoeuvring

Peng Wu

Mechanical Institute of Technology, Wuxi Institute of Technology, Wuxi, China and
School of Mechatronics Engineering and Automation, University of Shanghai, Shanghai, China

Shaorong Xie

School of Mechatronics Engineering and Automation, University of Shanghai, Shanghai, China

Hengli Liu

School of Mechatronics Engineering and Automation, University of Shanghai, Shanghai, China and
Mechanical Institute of Technology, Wuxi Institute of Technology, Wuxi, China, and

Ming Li, Hengyu Li, Yan Peng, Xiaomao Li and Jun Luo

School of Mechatronics Engineering and Automation, University of Shanghai, Shanghai, China

Abstract

Purpose – Autonomous obstacle avoidance is important in unmanned surface vehicle (USV) navigation. Although the result of obstacle detection is often inaccurate because of the inherent errors of LIDAR, conventional methods typically emphasize on a single obstacle-avoidance algorithm and neglect the limitation of sensors and safety in a local region. Conventional methods also fail in seamlessly integrating local and global obstacle avoidance algorithms. This paper aims to present a cooperative manoeuvring approach including both local and global obstacle avoidance.

Design/methodology/approach – The global algorithm used in our USV is the Artificial Potential Field-Ant Colony Optimization (APF-ACO) obstacle-avoidance algorithm, which plans a relative optimal path on the specified electronic map before the cruise of USV. The local algorithm is a multi-layer obstacle-avoidance framework based on a single LIDAR to present an efficient solution to USV path planning in the case of sensor errors and collision risks. When obstacles are within a layer, the USV uses a corresponding obstacle-avoidance algorithm. Then the USV moves towards the global direction according to fuzzy rules in the fuzzy layer.

Findings – The presented method offers a solution for obstacle avoidance in a complex environment. The USV follows the global trajectory planned by the APF-ACO algorithm. While, the USV can bypass current obstacle in the local region based on the multi-layer method effectively. This fact was validated by simulations and field trials.

Originality/value – The method presented in this paper takes advantage of algorithm integration that remedies errors of obstacle detection. Simulation and experiments were also conducted for performance evaluation.

Keywords Visibility graph, Artificial Potential Field-Ant Colony Optimization, Multi-layer obstacle-avoidance framework, Unmanned surface vehicle

Paper type Research paper

1. Introduction

Autonomous obstacle avoidance is an important task of an unmanned surface vehicle (USV), demanding algorithms for safely driving the USV towards a target without human intervention (Elkins *et al.*, 2010; Finn and Scheduling, 2010). Recently, research in autonomous global and local obstacle-avoidance algorithms has gained significant attention because of the increasing military and civilian needs for navigation in complex environments (Khansari-Zadeh and Billard, 2012). However, a remaining challenge yet is the

interaction of global and local obstacle avoidance. In existing local obstacle-avoidance approaches, artificial potential field (APF) is a classic obstacle-avoidance algorithm (Khatib, 1986; Pan *et al.*, 2015; Pang *et al.*, 2015). It is suitable for low-level on-line or off-line control of USV because of its explicit physical implication and simple mathematical description (Doitsidis *et al.*, 2011; Gago-Ben Tez *et al.*, 2014). However, there are several limitations, such as local minimum

The current issue and full text archive of this journal is available on Emerald Insight at: www.emeraldinsight.com/0143-991X.htm



Industrial Robot: An International Journal
44/1 (2017) 64–74
© Emerald Publishing Limited [ISSN 0143-991X]
[DOI 10.1108/IR-04-2016-0127]

This work was supported in part by the National Natural Science Foundation of China (No. 61233010, 61375093), the Nature Science Foundation of Shanghai (No.13ZR1454300), the National Natural Science Foundation for Distinguished Young Scholars (No.61525305). The authors would like to thank all the project partners for their valuable contribution.

Received 11 April 2016
Revised 18 September 2016
Accepted 22 September 2016

and goal non-reachable with obstacle nearby (GNRON) (Jaradat *et al.*, 2012; Kim and Lee, 2011; Lee *et al.*, 2012). Thus, a modified algorithm termed Bacterial Potential Field (BPF) was developed to achieve a feasible, optimal and safe trajectory, which improves the APF algorithm with the Bacterial Evolutionary Algorithm (BEA) (Krause *et al.*, 2013; Orozco-Rosas *et al.*, 2015). To optimize path and obtain the shortest safe path for ships, a method of finding optimal routes on raster planes was proposed (Hong and Murray, 2015; Szlapczynski, 2006). This method takes advantage of a new algorithm that tends to minimize a number of direction changes within a route, while steering clear of obstacles. To improve the precision of local obstacle avoidance, a high-precision navigation technique using dead-reckoning sensors and LIDARS is proposed for building a parameterized map of artificial bridge structures and estimating the vehicle's position relative to the parameterized map simultaneously. Field experiments were performed to verify and demonstrate the performance of the proposed navigation and mapping algorithms (Jungwook *et al.*, 2015). A lot of work was done on the improvements of local obstacle-avoidance algorithms. However, one algorithm cannot meet the needs, and multiple algorithms are used in local obstacle avoidance for different cases.

In global path planning, the evolutionary obstacle-avoidance algorithm was used for addressing these issues of the time needed (Tam and Bucknall, 2010a, 2010b). The study focused on the output consistency of such a stochastic-based algorithm, and the presented algorithm was shown to be effective by using constructed vector fields to direct the search routine towards the object regions. Additionally, another previous study generated modified obstacle-avoidance manoeuvring for the Ant Colony Optimization (ACO) algorithm. A fast two-stage ACO algorithm was presented based on the scent pervasion principle, and its basic theory splits heuristic search into two stages (preprocess stage and path-planning stage), and simulation shows that the algorithm performs well in the case of the high grid resolution (Chen *et al.*, 2013; Hong and Murray, 2015; Vinay and Sridharan, 2012). Particle Swarm Optimization (PSO) and ACO were also combined for eliminating the randomness of the classic ACO algorithm in the initial stage and for improving the rate of convergence (Dilmac *et al.*, 2014; Elloumi *et al.*, 2013, 2014). Otherwise, to solve the problem of obstacle avoidance in the complex environment including underwater obstacles, path re-planning techniques and underwater obstacle avoidance for USV based on multi-beam forward looking sonar (FLS) were presented, and near-optimal paths in static and dynamic environments with underwater obstacles are computed using a numerical solution procedure based on an A* algorithm. The performance of the proposed method was verified through simulations and sea experiments (Phanthong *et al.*, 2014). Overall, these existing algorithms are centred on global obstacle avoidance, and they are not able to control the USV to proceed in more complex environments than the environments with simple and few obstacles. Furthermore, most evasive manoeuvring in practice performed with a single algorithm do not take into account sensor errors.

This paper presents a type of cooperative manoeuvring for USV, and the complementary advantages of different

obstacle-avoidance methods is utilized. An off-line trajectory is generated by the execution of the Artificial Potential Field-ACO (APF-ACO) algorithm before the cruise of the USV, and the USV can track the path until appearance of unknown obstacles. To compensate for sensor errors, a type of multi-layer obstacle-avoidance framework is proposed for online obstacle avoidance in that case, while considering the mode of the USV's safe region, and the USV can return to the original off-line trajectory after bypassing the unknown obstacles.

The rest of this paper is organized as follows. Section 2 describes the model of safe space, and it takes into account local obstacle avoidance. Section 3 is the method of obstacle detection for eliminating errors in scanning data. Section 4 presents the global obstacle-avoidance algorithm that uses the APF-ACO algorithm for planning an off-line global trajectory before the operation of the USV. Section 5 established the multi-layer algorithm for local obstacle avoidance. Section 6 shows the simulation results and the field test results for verifying the presented algorithms. Section 7 is the conclusion.

2. Modelling of a safe space

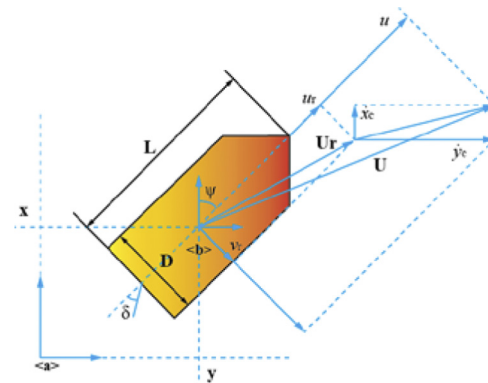
Assuming that the USV is moving in 2-D space, and neglecting pitch, roll and heave (Matos *et al.*, 2011; Sheng *et al.*, 2014), and an inertial, earth-fixed frame $\langle a \rangle$ and a body-fixed frame $\langle b \rangle$ are considered. Position and orientation of the USV in earth-fixed frame $\langle a \rangle$ are expressed as $[x \ y \ \psi]$. Surge and sway velocities, yaw rate and force and moments are denoted as $[u \ v]$ absolute and $[u_r \ v_r]$ with respect to the water, ω , $[X \ Y \ N]$ in a body-fixed frame $\langle b \rangle$, respectively. Figure 1 shows the hull model of a USV including reference frames, absolute and relative speed, actuator location and rudder angles. $[\dot{x}_c \ \dot{y}_c]$ denotes the sea current, and the body-fixed absolute velocity and velocity with respect to water are:

$$\begin{aligned} u &= u_r + \dot{x}_c \cos \psi + \dot{y}_c \sin \psi \\ v &= v_r - \dot{x}_c \sin \psi + \dot{y}_c \cos \psi \end{aligned} \quad (1)$$

The USV kinematics in the earth-fixed frame are:

$$\begin{aligned} \dot{x} &= u_r \cos \psi - v_r \sin \psi + \dot{x}_c \\ \dot{y} &= u_r \sin \psi + v_r \cos \psi + \dot{y}_c \\ \dot{\psi} &= \omega \end{aligned} \quad (2)$$

Figure 1 Hull model of a USV



In this work, we designed a method to calculate the safe space of a USV. As shown in Figure 2, the model of safe space is represented by four arcs, and the algorithm of this model can be used for calculating the safe space of the USV (Nishitani et al., 2015). In this study, to realize the collision-free motion of the USV, the proposed local path planning takes into consideration the USV's safe space. The main contribution of this study is that the robot can generate a safe robot-centred motion in a dynamic environment by taking cognizance of the future states of the robot. According to this method, the length of the long axis and short axis is defined as $O'A' = a$ and $O'C' = b$, respectively. The minimum length of a is:

$$a = \frac{1}{f} \cdot v + \frac{L}{2} \quad (3)$$

where v is the USV's velocity. The length of the short axis b is determined by a and the Arc 1 centred at the USV with the radius of b is derived. f denotes laser scanning frequency. The minimum length of b is:

$$b = 0.14a + \frac{D}{2} \quad (4)$$

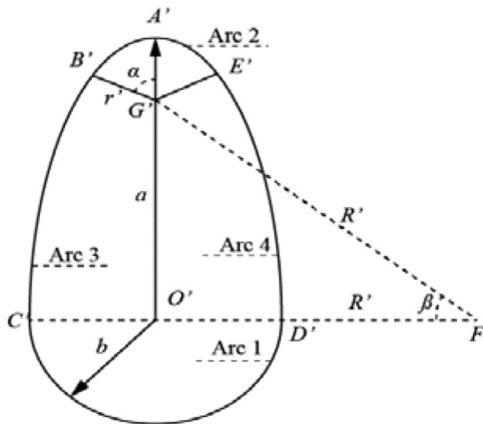
Arc 2 centred at G' with the radius of r' , and an arc chord angle of 2α is calculated by:

$$\begin{aligned} e &= \sqrt{a^2 + b^2} \\ d &= a - b \\ r' &= \frac{e(e - b)}{2a} \\ O'G' &= O'A' - r' = a - r' \\ \alpha &= \arctan\left(\frac{a}{b}\right) \end{aligned} \quad (5)$$

Where e is the chord length of Arc 2 and Arc 3, d is the difference of axis b and axis c , Arc 3 is centred at F' with the radius of R' , and a chord angle 2β is calculated by:

$$\begin{aligned} R' &= \frac{e(e + b)}{2a} \\ O'F' &= B'F' - b = R' - b \\ \beta &= \arctan\left(\frac{a}{b}\right) \end{aligned} \quad (6)$$

Figure 2 Model of a USV's safe space



Arc 4 is symmetric to the Arc 3. The safe space of USV is generated by combining of the above four arcs in Figure 2.

3. Obstacles detection

There exist errors in scanning data because of the effect of environment, for example, surface material reflectivity of obstacles, laser reflection direction and heavy weather can impact the effect of laser detection. Thus, it is necessary to filter original data for eliminating unreliable data and isolated data from scan results. In this work, we used median filtering to reduce noise. The fundament of median filter is to use mid-value in a neighbourhood of a point to replace the point. This method is capable of eliminating isolated noise (Omura et al., 2013; Shukla et al., 2014; Vijaykumar et al., 2014). The mid-value of the sample $X(n - N) \cdots X(n) \cdots X(n + N)$ is selected in the scanning area of LIDAR. Then, $Y(n) = \text{med}[X(n - N) \cdots X(n) \cdots X(n + N)]$.

Where $\text{med}[\]$ denotes selecting med-value after data rearrangement according to the order from large to small in the scanning range. The method of selecting med-value uses a certain 2-D sliding formwork to arrange data in the formwork according to data values. It generates a monotonic 2-D array of data.

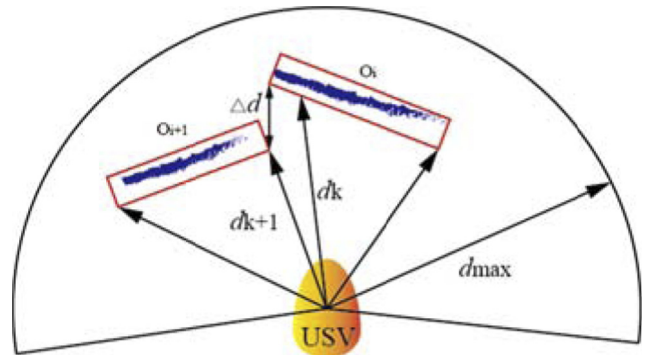
Median filtering of laser data uses 3×3 sliding window in consideration of time continuity and spatial continuity. Three points in the adjacent space and three frames data in the adjoining times can be selected, and medians of the nine points can be calculated as the distance value of the current point. The formula in 3×3 sliding window can be written as:

$$\begin{aligned} P(t, i) &= \text{Mid}(A) \\ A &= \{P(t - 1, i - 1), P(t - 1, i), P(t - 1, i + 1), \\ &\quad P(t, i - 1), P(t, i), P(t, i + 1), P(t + 1, i - 1), \\ &\quad P(t + 1, i), P(t + 1, i + 1)\} \end{aligned} \quad (7)$$

Where $\text{Mid}(A)$ represents the median of A , A denotes the laser data acquisition in the 3×3 sliding window.

LMS511 LIDAR was used for obstacles' detection in this work, with a scanning angle resolution of 0.5° and a scanned area of 190° . Its number of direction vectors is 361. In Figure 3, d_k is set as returned value of LIDAR in every vector direction, which denotes the distance between an obstacle point and a USV in any vector direction. d_{\max} represents the maximum effective distance in the scanning area. $\Delta d \approx |d_{k+1} - d_k|$ denotes the interval between two adjacent obstacles. The

Figure 3 Obstacle recognition



distance of adjacent points in the same point set is short; however, the distance of different point sets is long. Points in three frames are clustered, and different obstacles are identified according to this feature.

In Figure 3, d_k and d_{k+1} are not equal to d_{\max} , while they derive from o_i and o_{i+1} . Then, d_k and d_{k+1} are determined as one obstacle according to the threshold value Δd . When a USV passes the interval between two adjacent obstacles ($\Delta d > b$), o_i and o_{i+1} are determined as two different obstacles for selecting a feasible path. Otherwise, distinguishing different obstacles has no practical significance. Set $O = \{o_1, \dots, o_i, \dots, o_n\}$ denotes all obstacles in the scanning scope according to the rule of obstacle recognition.

4. Global obstacle-avoidance algorithm

4.1. Environment modelling

A USV is set in cells with binary information before the ACO algorithm is applied. The grids consist of obstacles and free regions. The binary information can be written as:

$$f(x) = \begin{cases} 0 \\ 1 \end{cases} \quad (8)$$

where $f(x) = 0$ represents the free grids, and $f(x) = 1$ denotes obstacle grids (Hu et al., 2013). The grid should be suitable for the size of the USV because it is easy to calculate the position of USV on the map.

4.2 Goal function

A USV moves in the working space established by the grids' model, and the length of the shortest path is:

$$\min L = \sum_{i=1}^T d(i) \quad (9)$$

$$P_i(x, y) \in \psi_f$$

where L denotes the length of the path, and $d(i)$ represents the distance of one step. $P_i(x, y)$ denotes the current position of the USV, and ψ_f represents a 2-D grid plane. When the USV is in a grid, it can select one of eight directions. $d(i)$ can be expressed as:

$$d(i) = \begin{cases} 1 & P_i(x) = P_{i+1}(x) \text{ or } P_i(y) = P_{i+1}(y) \\ \sqrt{2} & \text{else} \end{cases} \quad (10)$$

If $d(i)$ equals 1, the USV moves one unit in the horizontal or vertical direction, and if $d(i)$ equals $\sqrt{2}$, the USV moves $\sqrt{2}$ unit diagonally.

4.3 Search mechanism of APF-ACO

Dorigo et al. (1996) presented a classic bioinspired optimization algorithm termed ACO algorithm. The algorithm imitates the foraging behaviour of ants. Pheromone is used by the ACO algorithm as the basis of subsequent behaviours selected by ants, and every ant can respond to pheromone deposited by others within a certain region. Each ant can make probabilistic judgements for multiple paths according to pheromone strength. Thus, the foraging behaviour of ant colonies expresses a type of positive feedback

(i.e. more ants select a shorter path, and its pheromone strength becomes greater). The successor can select the greater possible path. In the ACO algorithm, the number of ant colonies is m , and an ant can select the next node according to:

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^{\alpha'} \cdot [\eta_{ij}(t)]^{\beta'}}{\sum_{j \in \mathcal{J}_k(i)} [\tau_{ij}(t)]^{\alpha'} \cdot [\eta_{ij}(t)]^{\beta'}}, & j \in \mathcal{J}_k(i) \\ 0, & \text{others} \end{cases} \quad (11)$$

where $\mathcal{J}_k(i)$ denotes a set of grids, and ant k can select it at the next step. The list $tabu_k$ records the grids, which is t passed by ant k . α' and β' are the relative importance degree of pheromone and directive factor respectively. η_{ij} denotes the comprehensive directive factor, which represents the expectation of transferring to the next grid and τ_{ij} is pheromone strength. The formulas for updating the pheromone after a loop are:

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij} \quad (12)$$

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k$$

where ρ represents the evaporation coefficient of pheromone, and $\Delta\tau_{ij}$ denotes the pheromone increment. The classic ACO algorithm has the problem of slow convergence and prematurity because of its inherent defects. Thus, the APF-ACO algorithm is used in our work.

The APF-ACO algorithm uses the resultant force of APF to change the transition rule. η_{ij} can be written as:

$$\eta_{ij}(t) = \eta_F(t) \cdot \eta_d(t) \quad (13)$$

where $\eta_d(t)$ represents the directive factor of distance, and $\eta_F(t)$ denotes the directive factor of potential force. $\eta_F(t)$ and $\eta_d(t)$ are:

$$\eta_d(t) = \frac{1}{d(P, T)}$$

$$\eta_F(t) = \lambda \cdot a^{F \cdot \cos\theta} \quad (14)$$

$$\lambda = \frac{K - K_m}{K}$$

where $d(P, T)$ denotes the shortest distance between the current waypoint and the goal, and λ represents the influence coefficient of a directive factor. K and K_m denote maximum iterations and the current iterations. a is a constant and $a > 1$. F denotes the potential resultant force, and θ represents the included angle between the potential resultant force and feasible path. Amounts of inferior solution generated and calculated amount are increased for lacking of pheromone and too small directive factor of distance at the initial stage. The directive factor of potential force plays a dominant role in the initial phase of APF-ACO algorithm, and the USV can search for the optimal path quickly. The effect of potential field is decreased with the increase of the iterations. The algorithm mainly depends on the influence coefficient of a directive factor gradually. This method is able to overcome the problem of premature convergence.

4.4 Updating pheromone

In the process of updating pheromone, the strategy based on distribution of wolves is applied. When wolves hunt, the following phenomenon has been found in previous studies. Most food caught by wolves is assigned to the stronger, but the puny are starved to death. Getting inspiration from this behaviour, the pheromone intensity of the optimal path is added up, while the pheromone intensity of the worst path is reduced, according to:

$$\begin{aligned}\tau_{ij}^{new} &= (1 - \rho)\tau_{ij}^{old} + \sum_{k=1}^m \Delta\tau_{ij}^k + \Delta\tau_{ij}^{best} - \Delta\tau_{ij}^{worst} \\ \Delta\tau_{ij}^{best} &= \begin{cases} \delta \cdot (Q/L^{best}) & (i,j) \in \text{optimal path} \\ 0 & \text{others} \end{cases} \\ \Delta\tau_{ij}^{worst} &= \begin{cases} \omega \cdot (Q/L^{worst}) & (i,j) \in \text{worst path} \\ 0 & \text{others} \end{cases}\end{aligned}\quad (15)$$

where L^{best} denotes the length of the local optimal path; L^{worst} is the length of the local worst path; Q is constant pheromone; and δ and ω represent the number of optimal and worst ants, respectively. This strategy is able to improve the ability of global optimal solution and the rate of convergence.

4.5 Potential field force in APF-ACO

The APF-ACO algorithm plans a path based on a directive factor of distance and artificial potential force, and the artificial potential force plays a dominant role in its infancy. Figure 4 shows the four potential cases of the APF-ACO algorithm on the grid-based map. In Figure 4(a), there is no obstacle between the current point and target, and the resultant potential force is equal to attraction. The included angle between resultant potential force and feasible path is smaller, the value of η_F is higher according to equation (13). Thus direction ① can be selected ($\theta_1 < \theta_2$). Likewise, direction ① of the next waypoint can also be selected based on

potential field force heuristic information in Figure 4(b)–(d). Convergence property of the APF-ACO algorithm can be further proved based on the above analysis (see Appendix). The Pseudo-Code of APF-ACO is shown in the following Pseudo-Code of APF-ACO algorithm:

- 1: Set parameters, initialize pheromone trails
- 2: While termination condition not met and there are no unexpected obstacles do
- 3: Construct ant solutions combined with heuristic information of APF
- 4: Apply local search
- 5: Update pheromones according to the distribution of wolves
- 6: End while
- 7: If unexpected obstacles are encountered, the current algorithm switches to Pseudo-Code 2

5. Local obstacle-avoidance algorithm

5.1 Local obstacle-avoidance layers

In the USV, the problem of how to avoid unknown obstacles quickly is important to be tackled. The relative position of the USV in regard to its environment and the direction of motion should be taken into consideration. We divide the measurement range of LIDAR into three layers, including emergency obstacle-avoidance layer, accurate obstacle-avoidance layer and fuzzy obstacle-avoidance layer.

5.2 Principle of the algorithms

When a USV is in the emergency obstacle-avoidance layer, rapid obstacle avoidance is necessary. Thus, emergency obstacle-avoidance algorithm should involve less computation. The region of emergency obstacle-avoidance layer is the intersection of the safe space and scanned area with LIDAR, according to the model of the USV. If there are unknown obstacles in the emergency obstacle-avoidance region, the USV moves along the opposite direction to the goal until the unknown obstacles is outside the emergency obstacle-avoidance region.

Visibility Graph (VG) method has extensively been studied as a fundamental path-planning method in the context of on-line obstacle-avoidance algorithms. It is directly used to search for the optimal path in the local region. VG was completely unknown initially, and it is incrementally constructed by integrating the local feedback information of the sensor during the navigation process, as illustrated in Figure 5. When a USV is in the accurate obstacle-avoidance layer, it is adopted.

When the position of the starting point or goal is changed, this algorithm can search for a new visibility point for planning a feasible path.

Firstly, whether the desired path is intersected with obstacles needs to be estimated. In Figure 5, obstacles are simplified as line segments. The two endpoints of expanded obstacles are $A(x_a, y_a)$ and $B(x_b, y_b)$ based on prior knowledge. The VG algorithm can decide whether line OG intersects with line AB .

The distance between Point A and Line OG is $d_a = (A'x_a + B'y_a + C')/\sqrt{A'^2 + B'^2}$.

The distance between Point B and Line OG is $d_b = (A'x_b + B'y_b + C')/\sqrt{A'^2 + B'^2}$.

If $d_a \cdot d_b \leq 0$, AB intersects OG . If $d_a \cdot d_b > 0$, AB does not intersect with OG .

Figure 4 Artificial potential field in grid environment

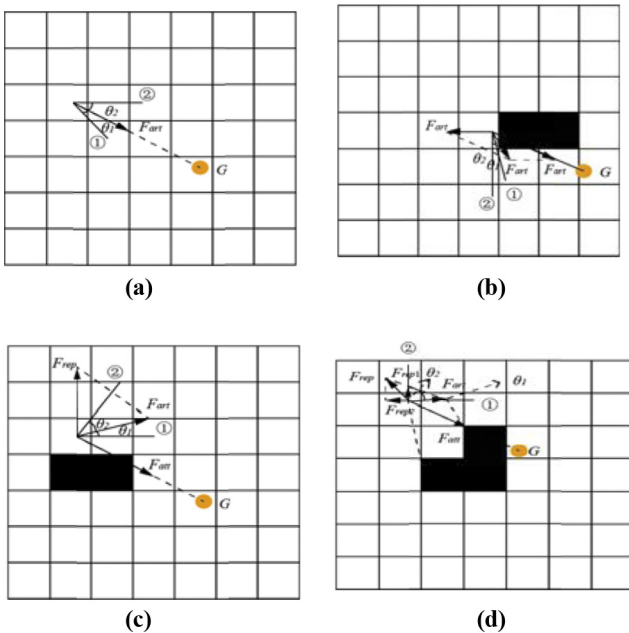
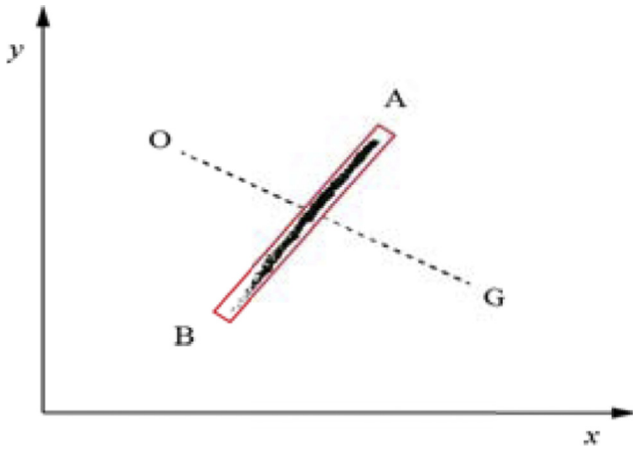


Figure 5 visibility graph obstacle-avoidance

Secondly, the VG algorithm seeks out a visible point between the position of USV and the goal. If there is no obstacle intersecting with expected trajectory, the visible point is the current position of USV, and USV moves along the expected trajectory. If there are obstacles intersecting with expected trajectory, the distances between the feature points of the current obstacle and the goal are calculated. The point belonging to a shorter distance is selected as the next visible point of the USV. The USV repeats the above path planning until reaching the vicinity of the target.

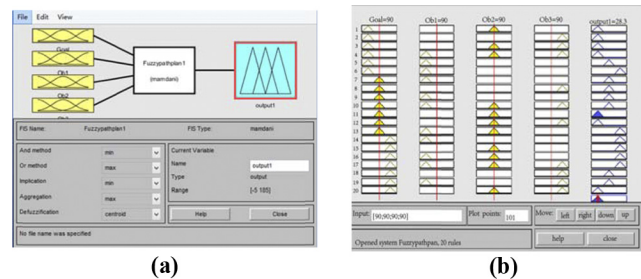
The measurement accuracy of LIDAR becomes poorer as the distance increases, and the USV is in the stage of fuzzy obstacle avoidance. The fuzzy obstacle-avoidance algorithm uses fast feedback to replace accurate calculation. This method determines the macroscopical direction of a USV based on the measured data of LIDAR. When the obstacles are in the accurate layer, the accurate algorithm is used again. Fuzzy obstacle-avoidance window is set up according to the effective measurement range. The region is divided into three subdomains, including front-left *FL*, front *F* and right-front *RF*. The designed fuzzy programming controller is described as follows.

The corresponding angle of *RF* is $[-5^\circ, 60^\circ]$. The corresponding angle of *F* is $[60^\circ, 120^\circ]$. The corresponding angle of *FL* is $[120^\circ, 185^\circ]$. Output variable *UD* denotes direction of USV. Its corresponding angle is $[-5^\circ, 185^\circ]$. Linear function is selected as the function definition. Figure 8 is the control structure of fuzzy obstacle avoidance, and 21 fuzzy control rules are summarized in Table I. *Ob1*, *Ob2* and *Ob3* represent the first, second and third obstacles, respectively. Expressions of fuzzy control rules is if (*Goal* is *x1* and *Ob1* is *x2* and *Ob2* is *x3* and *Ob3* is *x4*) then (*UD* is *y1*), and each input variable is selected in linguistic values.

The Mamdani method is used in fuzzy decision for fuzzy obstacle avoidance, and the Centre of Gravity Defuzzification is also adopted. As shown in Figure 6, simulation analysis is done by MATLAB fuzzy control toolbox based on the above fuzzy rules. As you can see in Figure 6, the USV can select direction based on fuzzy rules if there are obstacles in the fuzzy layer. Then the USV can continue to move until accessing the accurate layer again.

Table I Fuzzy rules

No.	Goal	Ob1	Ob2	Ob3	UD
1	RF	None	F	None	RF
2	RF	None	None	FL	RF
3	RF	None	F	FL	RF
4	RF	RF	F	FL	RF
5	RF	RF	None	None	F
6	RF	RF	None	None	FL
7	F	RF	None	None	F
8	F	None	None	FL	F
9	F	RF	None	None	F
10	F	RF	F	FL	F
11	F	None	F	None	RF
12	F	None	F	FL	RF
13	F	RF	F	None	FL
14	FL	RF	None	None	FL
15	FL	None	F	None	FL
16	FL	RF	F	None	FL
17	FL	RF	F	FL	FL
18	FL	None	None	FL	F
19	FL	RF	None	FL	F
20	FL	None	F	FL	RF

Figure 6 Fuzzy logic simulation

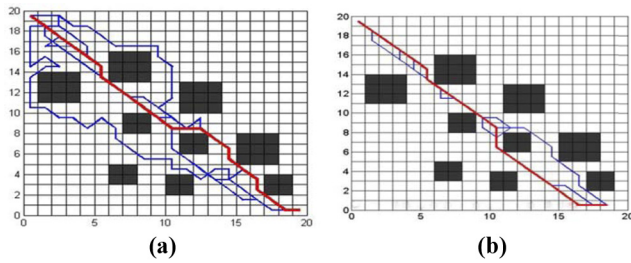
Notes: (a) Simulation interface; (b) Simulation result

6. Results and discussion

6.1 Simulation of APF-ACO

To verify the modified APF-ACO algorithm, the modified algorithm was simulated in a 20×20 grids model. In simulation, attraction gain was set to be $k = 20$, repulsion gain $\eta = 10$, initial step length $l^* = 0.1$, population size $m = 50$, the pheromone $\alpha' = 1$, relative importance degree of directive factor $\beta' = 10$, evaporative coefficient $\rho = 0.95$, additional strength of pheromone $Q = 1$.

Figure 7 is the simulated path of the two algorithms. In Figure 7(a) and (b), the red line is the shortest path of ACO algorithm, and the blue line is the other path. The path in Figure 7(b) is shorter than the one in Figure 7(a). In Figure 8, the red dashed line and blue dashed line denote the average path length of APF-ACO and ACO algorithm, respectively. The red solid line and the blue solid line denote the shortest path length of the APF-ACO and the ACO algorithm, respectively. Figure 8 shows that the convergence of the APF-ACO algorithm is faster than that of the ACO algorithm.

Figure 7 Simulative path of ACO and APF-ACO algorithms

Notes: (a) Trajectories of classic ACO; (b) Trajectories of APF-ACO

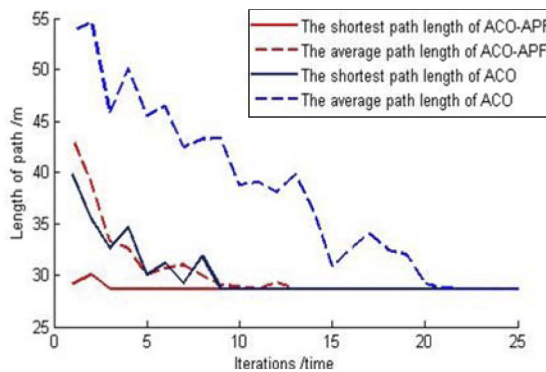
Figure 8 Convergence of APF-ACO and classic ACO algorithms

Table I shows the data of simulation result. The APF-ACO algorithm has less iterations and more accurate path length compared with the classic ACO algorithm. The calculating speed of the proposed method can be improved 38.1 per cent.

According to Table II, compared with the traditional ACO algorithm, the APF-ACO algorithm has fewer iterations of the shortest path length, fewer maximum iterations and more accurate path length. In summary, because of fewer iterations and faster convergence rate, the APF-ACO algorithm can drive the USV to reach a target rapidly via the relative optimal path.

In conclusion, because of less iterations and faster convergence rate, the APF-ACO algorithm can drive the robot to reach the target rapidly and seek the relative optimal path. The APF-ACO algorithm has a better effect of obstacle avoidance compared with the classic ACO. Thus, APF-ACO was adopted into the field test for planning a global path based on these simulation results.

6.2 Simulation of multi-layer algorithm

To validate the proposed multi-layer algorithm in local region, simulation was conducted based on the model of USV before the field test. The main parameters of the USV are summarized in Table III:

Table II Comparison of the simulation

Algorithm	Average path length	Shortest path length	Maximum iterations
APF-ACO	28.932	28.624	13
ACO	29.360	29.210	21

$$\omega(t) = \int_0^t \omega(\dot{t})d(t) = \int_0^t \frac{T(\alpha)}{J} d(t) \quad (16)$$

$$\alpha_{max} = \omega(t) \cdot \frac{n}{f}, n \geq 3$$

where $\omega(t)$ is the angular velocity of the USV, $T(\alpha)$ is torque of the USV and n is the laser scanning times.

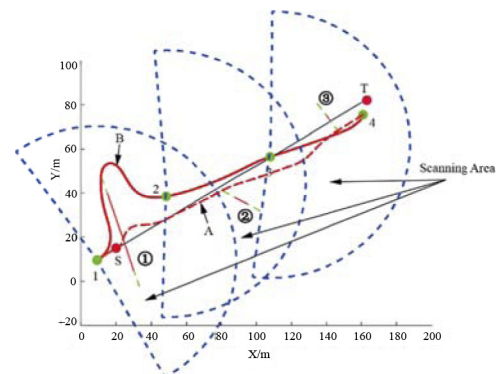
The maximum angle of steering gear α_{max} and calculated safe region of the USV must be considered before simulation, and the range of the LIDAR was divided into three parts. The proposed algorithm was then implemented into simulation.

In Figure 9, symbols ①, ②, ③ denote three obstacle clustering. S and T denote the starting point and target point, respectively. Points 1 (10, 15), 2 (45, 40), 3 (100, 50) and 4 (170, 70) represent the sub-goal points in each scanning area. To reduce computation, LIDAR data of the median filter were clustered as lines, and the lines were lengthened according to the size of the USV. The maximum range of LIDAR was set as the mobile scanning area for detecting the obstacles and target. A denotes the trajectory of the signal algorithm based on VG, and B represents the trajectory of multi-layer algorithm.

If the USV moves along Trajectory A which is intersected with symbol ①, it can pass through symbol ① and reach Point 4 nearby target T, and Trajectory A is not suitable for the USV because of safety. When the USV moves along Trajectory B from S point and symbol ① is in the emergency layer, it moves towards the opposite direction of T and reaches 1. Then symbol ① is in the accurate layer, and the VG algorithm controls the USV for avoiding symbol ① and reaching 2. Then the USV moves toward T according to the fuzzy rules. If it detects symbol ① or symbol ③, it also can avoid the obstacles and reach the sub-goal in the current area with the appropriate algorithm of the multi-layer

Table III Main parameters of the USV used in this work

Size	6.28m × 2.22 m
Speed	10 Kn
Heading angle	−30°
Rudder angle α	≤30°
Laser scanning frequency f	25 Hz
Maximum range of LIDAR	80 m
Precise range of LIDAR	50 m

Figure 9 Comparison of the single algorithm and multi-layer algorithm

algorithm. The length of Trajectories A and B can be calculated based on Figure 9. Although A is shorter than B by 20 per cent, the simulation results obviously show that the presented multi-layer algorithm can improve the safety of obstacle avoidance compared with adopting a single algorithm. Besides comparing the safety of path, the time for traveling from S to T which was obtained by running on an Intel(R) Core i3-3220 CPU 3.3 GHz, 2G RAM processor machine and the traveling times of A and B were 0.0423 and 0.0327 s. The time of B can be improved 22.7 per cent compared with A, and the path-planning with multi-layer algorithms framework is more efficient than a single algorithm. Thus, the multi-layer algorithm can be applied in the field test based on the analysis above.

6.3 Field test

To verify the presented method, field test was done for USV in a certain region. Figure 10 shows the USV of the field test, and its main structure parameters are written in Table. Figure 11 represents hardware system configuration of obstacle avoidance, which consists of obstacle detection, local obstacle avoidance and global obstacle avoidance. As known from Figure 11, obstacle detection was the precondition for the performance of the multi-layer algorithm, and its main hardware was GPS antenna, GPS receiver, LIDAR and INS. The APF-ACO algorithm was used for the global obstacle avoidance to calculate a safe ship route, and it is also a short voyage distance.

Then the USV performed the task of local obstacle avoidance with the multi-layer algorithm. As shown in Figure 12, the first process of global path planning (yellow trajectory) was performed off-line with the APF-ACO algorithm, and the USV followed the global path until encountering unknown obstacles in the certain region (blue circle). Then three field trials were done for verifying the application effect of the presented local algorithm in the local region.

In Figure 13, two obstacles were set in the local water area, where static obstacle ① and ② were set in the emergency layer and accurate layer, respectively. When the USV is navigated into the local water area in Figure 10, two modules of local obstacle-avoidance system were used for detecting obstacles for processing data with median filtering, and it clustered the



Figure 10 USV in the field trials

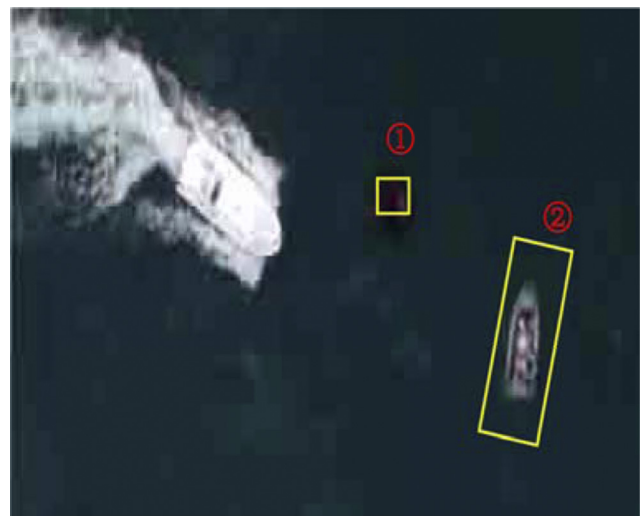
Figure 11 Hardware system configuration of obstacle avoidance



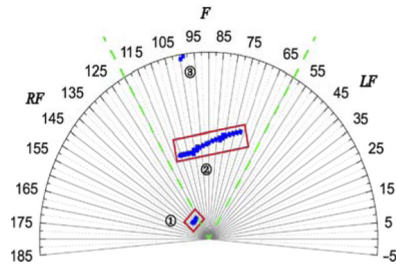
Figure 12 The water area of field trials



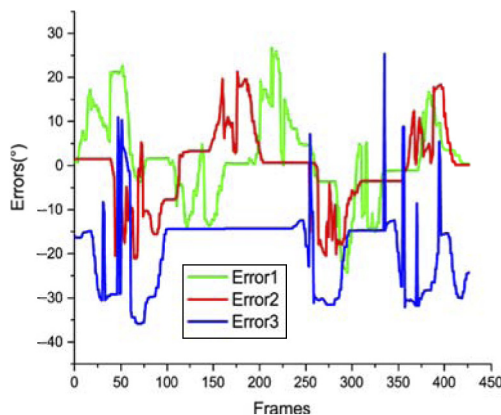
Figure 13 Obstacles in the local water area



LIDAR data in the local region as shown in Figure 14. As shown in Figure 14, LIDAR points' clouds can be clustered as rectangle (symbol ① and symbol ②) in emergency and accurate layer, respectively, and symbol ③ denoted some data in the F region of fuzzy layer can be detected according to prior knowledge. The USV executed the presented multi-layer algorithm for bypassing the current obstacles, and the effect of fired trial is shown in

Figure 14 The result of obstacle detection**Figure 15.** As shown in Figure 15, the USV bypassed the obstacles with the emergency algorithm and the accurate algorithm in the multi-layer framework successfully.

As shown in Figure 16, with the comparison of the actual rudder angle and order rudder angle, the errors of three tests are 17.09° – 19.49° (Error 1), -15.79° to 13.87° (Error 2), -36.17° to 5.62° (Error 3). The main reason for errors of rudder angle is environmental disturbance according to the model analysis. However, errors are in the safe range of $\pm 40^\circ$, and the USV can bypass the obstacle in Figure 15. Thus, it meets the safety requirements in this case, and the presented algorithm is feasible.

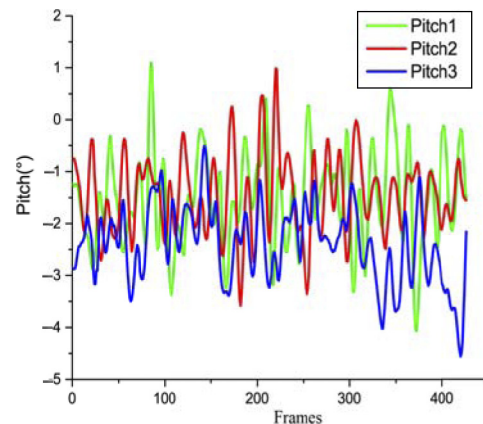
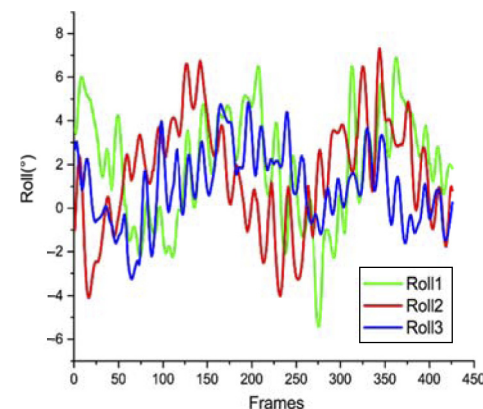
Figure 15 The test result of local obstacle avoidance**Figure 16** The error curve of rudder angle

Pitch and roll of the USV can be generated by environment disturbances during the process of the local obstacle avoidance. Figures 17 and 18 represent the curves of pitch and roll angle of the USV in the three trials. As known from Figure 16, the ranges of pitch angle are -3.71° to 0.58° , -3.42° to 0.51° , -4.32° to 0.48° . As shown in Figure 18, the ranges of roll angle are -5.49° to 6.85° , -4.15° to 7.37° , -3.19° to 4.81° which are within the safety range (the pitch angle is below $\pm 8^\circ$ and the roll angle is below $\pm 10^\circ$). Thus, the presented algorithm adopted in the field trials cannot affect the navigation safety of the USV, and it is practical.

7. Conclusion

We presented an obstacle-avoidance system that enables a USV to autonomously navigate through the region with obstacles in this paper.

A novel cooperative manoeuvring approach was adopted by the system. Global navigation with the APF-ACO algorithm was performed in a sea area before the movement of USV based on this method. The USV followed the global trajectory according to the steering instruction until encountering unknown obstacles in the local region on the map. It then avoided the obstacles with the proposed multi-layer algorithm after processing LIDAR data. This approach was also simulated for verifying the advantages of the APF-ACO algorithm and the multi-layer method proposed in this paper.

Figure 17 Variable curves of pitch angle**Figure 18** Variable curves of roll angle

The novel cooperative manoeuvring approach of the navigation system was implemented in the field test, and the results validated the proposed approach. In the future, it is necessary for reducing the environmental disturbance during the process of obstacle avoidance and improving the precision. We believe that the cooperative manoeuvring approach is the fundamental prerequisite for USVs offering services for obstacle avoidance under complex marine conditions.

References

- Chen, X., Kong, Y., Fang, X. and Wu, Q. (2013), "A fast two-stage ACO algorithm for robotic path planning", *Neural Computing and Applications*, Vol. 22 No. 3, pp. 313-319.
- Dilmac, S., Nizam, A. and Korurek, M. (2014), "Evaluation of a new heart beat classification method based on ABC algorithm, comparison with GA, PSO and ACO classifiers", *International Journal of Reasoning-based Intelligent Systems*, Vol. 6 No. 4, pp. 98-108.
- Doitsidis, L., Renzaglia, A., Weiss, S., Kosmatopoulos, E., Scaramuzza, D. and Siegwart, R. (2011), "3D Surveillance coverage using maps extracted by a monocular slam algorithm", *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1661-1667.
- Dorigo, M., Maniezzo, V. and Coloni, A. (1996), "Ant System: optimization by a colony of cooperating agents", *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, Vol. 26 No. 1, pp. 29-41.
- Elkins, L., Sellers, D. and Monach, W.R. (2010), "The Autonomous Maritime Navigation (AMN) project: field tests, autonomous and cooperative behaviors, data fusion, sensors, and vehicles", *Journal of Field Robotics*, Vol. 27 No. 6, pp. 790-818.
- Elloumi, W., Baklouti, N., Abraham, A. and Alimi, A.M. (2013), "Hybridization of Fuzzy PSO and Fuzzy ACO applied to TSP", *2013 13th International Conference on Hybrid Intelligent Systems (HIS)*, pp. 105-110.
- Elloumi, W., El abed, H., Abraham, A. and Alimi, A.M. (2014), "A comparative study of the improvement of performance using a PSO modified by ACO applied to TSP", *Applied Soft Computing*, Vol. 25 No. C, pp. 234-241.
- Finn, A. and Scheduling, S. (2010), "Developments and challenges for autonomous unmanned vehicles", Vol. 3, pp. 1-11.
- Gago-Ben Tez, A., Fern Ndez-Madrigal, J.A. and Cruz-Mart, N.A. (2014), "Marginal probabilistic modeling of the delays in the sensory data transmission of networked telerobots", *Sensors*, Vol. 14 No. 14, pp. 2305-2349.
- Hong, I. and Murray, A.T. (2015), "Assessing raster GIS approximation for euclidean shortest path routing", *Transactions in GIS*, Vol. 19 No. 5, pp. 499-515.
- Hu, S., Chen, C., Zhang, A., Sun, W. and Zhu, L. (2013), "A small and lightweight autonomous laser mapping system without GPS", *Journal of Field Robotics*, Vol. 30 No. 5, pp. 784-802.
- Jaradat, M.A.K., Garibeh, M.H. and Feilat, E.A. (2012), "Autonomous mobile robot dynamic motion planning using hybrid fuzzy potential field", *Soft Computing*, Vol. 16 No. 1, pp. 153-164.
- Jungwook, H., Jeonghong, P., Taeyun, K. and Jinwhan (2015), "Kim precision navigation and mapping under bridges with an unmanned surface vehicle", *Autonomous Robots*, Vol. 38 No. 4, pp. 349-362.
- Khansari-Zadeh, S.M. and Billard, A. (2012), "A dynamical system approach to realtime obstacle avoidance", *Autonomous Robots*, Vol. 32 No. 4, pp. 433-454.
- Khatib, O. (1986), "Real-time obstacle avoidance for manipulators and mobile robots", *The International Journal of Robotics Research*, Vol. 5 No. 1, pp. 90-98.
- Kim, J. and Lee, B.H. (2011), "Non-oscillatory multi-robot motion for stable target capture", *International Journal of Robotics and Automation*, Vol. 26 No. 4, pp. 441-449.
- Krause, J., Cordeiro, J., Parpinelli, R.S. and Lopes, H.S. (2013), "A survey of swarm algorithms applied to discrete optimization problems", *Swarm Intelligence and Bio-Inspired Computation: Theory and Applications: Elsevier Science & Technology Books*, Vol. 4 No. 9, pp. 169-191.
- Lee, J., Nam, Y., Hong, S. and Cho, W. (2012), "New potential functions with random force algorithms using potential field method", *Journal of Intelligent & Robotic Systems*, Vol. 66 No. 3, pp. 303-319.
- Matos, V., Simos, A. and Sphaier, S. (2011), "Second-order resonant heave, roll and pitch motions of a deep-draft semi-submersible: theoretical and experimental results", *Ocean Engineering*, Vol. 38 Nos 17/18, pp. 2227-2243.
- Nishitani, I., Matsumura, T., Ozawa, M., Yoroze, A. and Takahashi, M. (2015), "Human-centered X-Y-T space path planning for mobile robot in dynamic environments", *Robotics and Autonomous Systems*, Vol. 66, pp. 18-26.
- Omura, Y., Masuda, H. and Mimura, Y. (2013), "Feasibility investigation of obstacle-avoiding sensors unit without image processing", *Journal of Sensors*, Vol. 2013 No. 1, pp. 1-21.
- Orozco-Rosas, U., Montiel-Ross, O.H. and Sepulveda-Cruz, R. (2015), "Pseudo-bacterial potential field based path planner for autonomous mobile robot navigation", *International Journal of Advanced Robotic Systems*, Vol. 12 No. 81, pp. 1-14.
- Pan, Z., Li, J.Q., Hu, K.M. and Zhu, H. (2015), "Intelligent vehicle path planning based on improved artificial potential field method", *Applied Mechanics and Materials*, Vol. 742 No. 1, pp. 349-354.
- Pang, M., Meng, Z., Zhang, W. and Ru, C. (2015), "MGRO recognition algorithm-based artificial potential field for mobile robot navigation", *Journal of Sensors*, Vol. 501, p. 283693.
- Phanthong, T., Maki, T., Ura, T., Sakamaki, T. and Aiyarak, P. (2014), "Application of A* algorithm for real-time path re-planning of an unmanned surface vehicle avoiding underwater obstacles", *Journal of Marine Science & Application*, Vol. 13 No. 1, pp. 105-116.
- Sheng, G., Yan-Feng, W., Jiu-Cai, J. and Zhao-Hui, H. (2014), "A marine monitoring two-body unmanned surface vehicle motion modeling and validation studies", *2014 11th World Congress on Intelligent Control and Automation (WCICA)*, pp. 3118-3121.
- Shukla, A.K., Bhateja, V., Verma, R. and Alam, M.S. (2014), "An improved directional weighted median filter for restoration of images corrupted with high density impulse noise", *2014 International Conference on Optimization, Reliability, and Information Technology (ICROIT)*, pp. 506-511.

- Szlapczynski, R. (2006), "A new method of ship routing on raster grids, with turn penalties and collision avoidance", *Journal of Navigation*, Vol. 59 No. 1, pp. 27-42.
- Tam, C. and Bucknall, R. (2010a), "Collision risk assessment for ships", *Journal of Marine Science & Technology*, Vol. 68 No. 12, pp. 1145-1155.
- Tam, C. and Bucknall, R. (2010b), "Path planning algorithm for ships in close range encounters", *Journal of Marine Science & Technology*, Vol. 15 No. 4, pp. 395-407.
- Vijaykumar, V., Mari, G.S. and Ebenezer, D. (2014), "Fast switching based median-mean filter for high density salt and pepper noise removal", *AEU-International Journal of Electronics and Communications*, Vol. 68 No. 12, pp. 1145-1155.
- Vinay, V. and Sridharan, R. (2012), "Development and analysis of heuristic algorithms for a two-stage supply chain allocation problem with a fixed transportation cost", *International Journal of Services and Operations Management*, Vol. 12 No. 12, pp. 244-268.

Further reading

- Simmons, R. (1996), "The curvature-velocity method for local obstacle avoidance", *1996 IEEE International Conference on Robotics and Automation*, pp. 3375-3382.

Appendix. Convergence proofs of APF-ACO

The first theoretical aspect of ACO that we consider in this paper is the convergence problem. ACO algorithms are stochastic search procedures in which the bias due to the pheromone trails could prevent them from ever reaching the optimum. It is important to note that, when considering a stochastic optimization algorithm, there are two types of convergence: convergence in value and convergence in solution. Finding the optimal solution in optimization is the central problem. Once it is solved and the algorithm can be stopped.

First, $ACO_{bs, \tau_{min}}$ can be defined, and its convergence in value with probability 1 is proved. Then we show that the proof of convergence in value applies directly to the ACO-APF algorithm present in this paper. $ACO_{bs, \tau_{min}}$ is defined as follows. First in the Ant solution construction procedure, the initial location of each ant is chosen in a problem-specific way, and the dependence on the heuristic η . $F_{ij}(\tau_{ij}) = \tau_{ij}$ is assumed to be of the form used in all ACO algorithms, where $0 < \alpha < +\infty$ is a parameter. The probabilistic construction rule of equation (10) to build solutions becomes:

$$P_T(c_{h+1} = j | x_h) = \begin{cases} \frac{F_{ij}(\tau_{ij})}{\sum_{(i,j) \in N_i^k} F_{ij}(\tau_{ij})}, & \text{if } (i,j) \in N_i^k; \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

Second, the pheromone update procedure is implemented by choosing $\hat{S}_0 = s^{bs}$, and a lower limit $\tau_{min} > 0$ is put on the value of pheromone trails. The value of τ_{min} is a parameter of

$ACO_{bs, \tau_{min}}$, and we assume that $\tau_{min} < q(s^*)$. In this section, we prove that $ACO_{bs, \tau_{min}}$ is guaranteed to find an optimal solution with a probability that can be made arbitrarily close to 1 if given enough time (convergence in value).

Theorem 1 Let $P^*(\theta)$ be the probability that the algorithm that finds an optimal solution at least once within the first θ iterations. Then, for an arbitrarily small $\varepsilon > 0$ and for a sufficiently large θ it holds that: $P^*(\theta) \geq 1 - \varepsilon$,

and, by definition, $\lim_{\theta \rightarrow \infty} P^*(\theta) = 1$

Proof Due to the pheromone trail limits τ_{min} and τ_{max} we can guarantee that any feasible choice in equation (15) for any partial solution x_h is made with a probability $p_{min} > 0$. A trivial lower bound for p_{min} is given by:

$$p_{min} \geq \hat{p}_{min} = \frac{\tau_{min}^\alpha}{(N_c - 1)\tau_{max}^\alpha + \tau_{min}^\alpha},$$

Where N_c is the cardinality of the Set C of components. (For the derivation of this bound we consider the following "worst-case" situation: the pheromone trail associated with the desired decision is τ_{min} , while all the other feasible choices – there are at most $N_c - 1$ – have an associated pheromone trail τ_{max} .) Then, any generic solution s , including any optimal solution $s^* \in S^*$, can be generated with a probability $p_{min} \geq \hat{p}_{min} > 0$, where $n < +\infty$ is the maximum length of a sequence. Because it is sufficient that one ant finds an optimal solution, a lower bound for $P^*(\theta)$ is given by:

$$\hat{P}^*(\theta) = 1 - (1 - \hat{p})^\theta.$$

By choosing a sufficient large θ , this probability can be made larger than any value $1 - \varepsilon$. Hence, we have that $\lim_{\theta \rightarrow \infty} \hat{P}^*(\theta) = 1$.

Then incorporating such heuristic information into $ACO_{bs, \tau_{min}}$. The most common choice is $F_{ij}(\tau_{ij}) = [\tau_{ij}]^\alpha [\eta_{ij}]^\beta$, and $\eta_{ij} = \eta_F(t) \cdot \eta_d(t)$ is the heuristic factor in the ACO-APF algorithm. In this case equation (15), becomes:

$$P_T(c_{h+1} = j | x_h) = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{(i,j) \in N_i^k} [\tau_{ij}]^\alpha [\eta_{ij}]^\beta}, & \text{if } (i,j) \in N_i^k; \\ 0, & \text{otherwise} \end{cases} \quad (18)$$

where η_{ij} measures the heuristic desirability of ACO-APF. In fact, Theorem 1 can never be affected by the heuristic information; if we have $0 < \eta_{ij} < +\infty$ for each $(i,j) \in L$ and $\beta < \infty$. In fact, with these assumptions η is limited to some (instance-specific) interval $[\eta_{min}, \eta_{max}]$, with $\eta_{min} > 0$ and $\eta_{max} < +\infty$. Then, the heuristic information has only the effect of changing the lower bounds on the probability p_{min} of making a specific decision (the analogous estimates in the proofs of Theorem 1).

Corresponding author

Shaorong Xie can be contacted at: srxie@shu.edu.cn