8th International Conference on Information Science and Technology
June 30-July 6, 2018; Granada, Cordoba, and Seville, Spain

A Hybrid Path-Planning Scheme for an Unmanned Surface Vehicle

Ning Wang, Yuncheng Gao, Zhongjiu Zheng, and Hong Zhao
Center for Intelligent Marine Vehicles
School of Marine Electrical Engineering
Dalian Maritime University
Dalian, 116026, China
n.wang.dmu.cn@gmail.com, gyc3333@163.com

Jianchuan Yin
Center for Intelligent Marine Vehicles
Navigation College
Dalian Maritime University
Dalian, 116026, China
yinjianchuan@dlmu.edu.cn

***Abstract*—In this paper, a hybrid path-planning scheme that combines global A* algorithm with local dynamic window based collision avoidance is proposed for an unmanned surface vehicle (USV) in complex environment. Considering the way-points tracking in maritime applications, the A* algorithm with post-smoothed (A* PS) method is employed to reduce the number of way-points, and thereby contributing to plan a shortest path without constraining on grids. The local collision avoidance is realized by the Dynamic Window approach which takes the motion dynamics of the USV into account. Furthermore, a virtual safety zone pertaining to the shape of the obstacle is established to ensure reliable navigation at high speed. Simulation studies demonstrate that the proposed global-local hybrid path-planning scheme achieves remarkable performance and superiority in path planning with obstacle avoidance.**

***Keywords*—hybrid path planning; obstacle avoidance; A* PS; dynamic window; virtual safety zone; USV**

I. Introduction

It is well recognized that the uses of unmanned surface vehicle in the field of civilian and military have become increasingly vital in recent years. In this context, there have been many advanced theoretical achievements and practical applications on this issue [1]–[5]. Autonomous obstacle avoidance, which includes the global path planning and the local obstacle-avoidance in unknown environments [6], becomes the challenge due to driving the USV towards a destination without manual operation. With completely known environmental information, a optimal path can be calculated from the starting point to the goal off-line by global path planning algorithm. In addition, local path planning algorithm can calculate a viable path online according to the real-time environmental information obtained by sensors.

Among the numerous global path planning algorithms, a grid map-based path which can discrete continuous terrain is one of the best known methods in robotics and video games [7], [8]. Tracing the source, the grid map-based algorithm stems from [9] in which calculated the shortest path between two given nodes. Then the A-Star (A*) algorithm, which used heuristics to find the optimal path, was proposed based on grids in [10]. The main drawback of this approach is that the path must be in a particular angle between the preceding and following nodes, in other words, the path is constrained to be formed by the edges or vertices of grids. On the other hand, in order to satisfy precision to the planning path, the map needs to be separated into grids intensively, and thereby generating a large number cells on the path. However, the USV used in maritime usually performs the task relying on way-points tracking [11] that expects fewer way-points for easy implementation. The Dynamic A* (D*) algorithm, a version of A*, finds the optimal trajectory in real-time by incrementally replanning paths to the vehicle's state as environment is changed [12]. Subsequently, A* on visibility graphs (VG) [13] without constraining the path to grid edges or vertices, is employed to find true shortest path. VG interleaves searching each visibility graph edge, whose number can grow quadratically, thereby leading to reducing the efficiency of the system because of more running time than the A* algorithm.

As important as the former subject, the local obstacle-avoidance in unknown environments plays a primary role in navigation of USV. Artificial potential field (APF) is a well-known algorithm that the intelligent vehicle motion is determined by a virtual force combined by "Gravity" and "Repulsion" [14], [15]. It should be noted that the limitations of employing APF are local minimum and goal non-reachable. Considering the local minimum, the Tangent-Bug algorithm belonging to classical bug algorithms is introduced in [16]–[18], and the performance between Tangent-Bug and other bug is also compared in these documents. There are two main modes in classical bug algorithms: moving toward the target and moving along an obstacle boundary. Tangent-Bug can make a transition according to a globally convergent criterion which is determined by the distance from the robot to the goal. In [19], the version vector field histogram (VFH+) method is proposed for further improvement virtual force field histogram and vector field histogram. The method is for real-time local obstacle avoidance using an occupancy grid representation for obstacles information, then the robot's motion direction and velocity are calculated by this occupancy grid. In addition, the contribution of this method is taking the robot's size into consideration. However, most of these local algorithms do not consider the motion dynamics, thereby resulting in an infeasible path. In other words, the trajectories generated by those approaches could be followed if only infinite forces can asserted on the robot. But in the actual situation, the velocities can not be reached within a time interval due to the limitation of the robot's acceleration.

Motivated by the above observations, both global and local algorithms have their disadvantages. In this paper, a hybrid approach is proposed by combining a global path planning using A* post-smooth algorithm based on way-points tracking with a local obstacle-avoidance $Dynamics\ Window$ [20] algorithm that takes USV dynamics into account.

II. Global Path Planning

A. A* Algorithm

A-star method, or simply A*, is a basic path planning algorithm in various fields, which uses heuristics to find the optimal path. A* algorithm is correct, complete and optimal. The meaning of correctness is that there are no obstacles on the computational path. Completeness means that a path can be found , if feasible path exists. Optimality means that it is able to calculate the shortest path from the start point to the goal point.

Define an evaluation function for the A*

$$f(n) = g(n) + h(n) \quad (1)$$

where

- $g(n)$ is the cost of the path from the start point to node n found so far. $g(n)$ is given by

$$g(n) = g(n-1) + c(n, n-1) \quad (2)$$

  where $c(n, n-1)$ is the cost from node $n-1$ to its neighbor node n that calculated by using the Euclidean distance.
- $h(n)$ is the straight-line distance from the node n to the goal point. For predefined goal piont, the value of $h(n)$ is constant which can be calculated in advance.
- $f(n)$ is an estimate of the cost of an optimal path from start point via node n to the goal point.

There are two important priority queues in the A* algorithm, open list and close list. $Open$ is a stable priority queue, which storages nodes with the key f value in ascending order. $Close$ storages expanded nodes to ensure that each node is only expended once. The lowest f value in the $Open$ will be selected as he current node, and the path is found when the current node is goal point.

B. A* RCN

In order to apply A* algorithm to the USV, continuous maritime map needs to be discretized into grids. Grids are data structures in which all the cells are labeled existing obstacles or not. The precision of path and avoiding obstacles determined by the number of cells. In marine practice, the USV guidance path usually consist of way-points. Thus, the cells generated by A* are the way-points on guidance path. To satisfy the precision of the path, a large number of way-points may be generated. However, carrying out so many way-points mission can slow down the application significantly. Thus, it is vital to reduce the number of way-points under the condition of ensuring the accuracy.

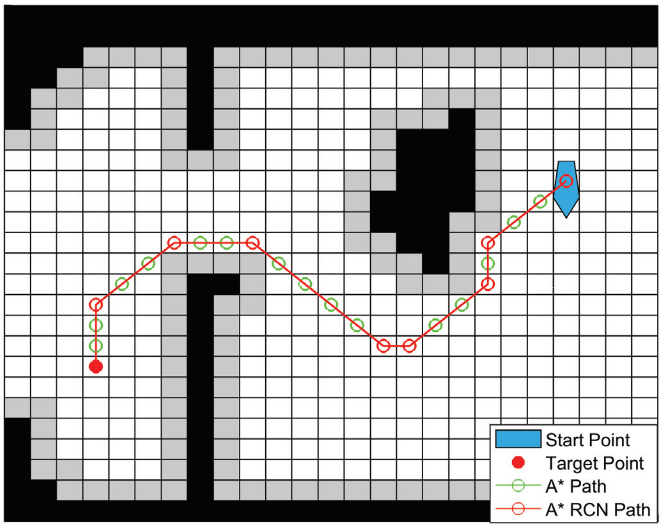


Fig. 1. A* RCN.

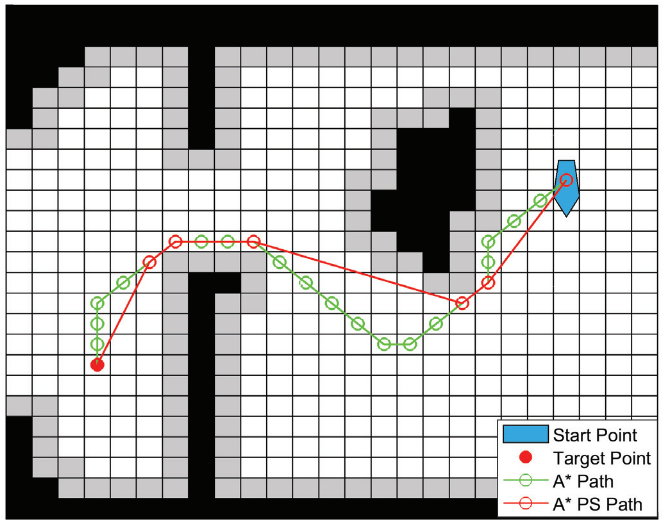


Fig. 2. A* PS.

One strategy can solve this problem is removing collinear nodes (A* RCN). Fig. 1 shows the example of the regulation that reduce many way-points after obtaining the original A* path, in which the gray grid represents the obstacle expansion area. The improvement algorithm checks whether adjacent three nodes in a straight line from the start point. If so, improvement algorithm removes the middle node and leaves the end-nodes as way-points. Improvement algorithm then repeats this procedure by checking again whether the nodes in a straight line.

C. A* PS

The nodes motion was constrained by the path generated by A* on grid, which must be in a particular angle between the preceding and following nodes. Therefore the path is lower the cost on grid but is not the shortest distance in reality.

A* with post-smoothed paths (A* PS) is appropriate for solving the USV path planning problem, which not only reduce the number of way-points but also can find shorter path. A* PS is the result of processing the A* with a short longer time, and **Algorithm 1** shows the procedure of it. The input is A* path $[s_0, ..., s_n]$, s_0 and s_n, represent the start point and the end point, respectively. Firstly, it checks whether the start point has line-of-sight to the third node. If not, it reserves the middle node. **Algorithm 2** shows the Line-of-sight algorithm whose function is checking whether there are obstacles between two nodes or not. Line-of-sight is the core of the algorithm that eliminates the limitation of motion in the grid such that shortens the path. Fig. 2 demonstrates the effect of applying A* PS that shortens the path. It is clear that the way-points generated by A* PS is much less than A*, even less than A* RCN.

Algorithm 1: A* PS

Input: A* path $[s_0, ..., s_n]$
Output: Optimized path
$j = 0$;
$l_j = s_0$;
for $i = 1...n-1$ **do**
  if $NOT\ LineOfSight(l_j, s_{(i+1)})$ **then**
    $j = j + 1$;
    $l_j = s_i$;
$j = j + 1$;
$l_j = s_n$;
return $[l_0, ...l_j]$;

Algorithm 2: Line of Sight

Input: two nodes to be detected [FromNode,ToNode]
Output: whether visible
for $each\ FromNode, ToNode$ **do**
  if $obstacles\ on\ connection\ line$ **then**
    return $true$;
  else
    return $false$;

III. Local Collision Avoidance

A. The Dynamic Window

$Dynamic\ Window$ takes the kinematic performance of the USV into account , giving a local collision avoidance method that calculates only the linear velocity and angular velocity that the vessel can reach in a given time. It is assumed that the velocity and angular velocity are constant within a given time-interval, thus the vessel travels along arcs.

One of the limitation of method is satisfying the admissible velocities (u_a, r_a), where u_a is linear velocity and r_a is yaw rate. The purpose of limitation is that the vessel can stop urgently before encountering an obstacle. The admissible velocities is given by

$$V_a = \{u, r \big| |u| \leq \sqrt{2 \cdot dist(u,r) \cdot \dot{u}_b}, \\ |r| \leq \sqrt{2 \cdot dist(u,r) \cdot \dot{r}_{max}}\} \quad (3)$$

where $dist(u, r)$ represents the distance to the closest obstacle that intersects with arc trajectories.. $\dot{r}_{max}$ is the maximum breaking angular acceleration, and u_b is the maximum acceleration when stopping, which is defined as

$$\dot{u_b} = \begin{cases} \dot{u}_{min}, & u \geq 0 \\ \dot{u}_{max}, & u < 0 \end{cases} \quad (4)$$

where $\dot{u}_{max} > 0$ is the maximum surge acceleration of the vessel, and $\dot{u}_{min} < 0$ is the maximum reversing acceleration.

The second limitation of method is satisfying a dynamic window in which the velocities can be reached within the next time interval. Based on the current velocities of the vessel (u_a, r_a), the dynamic window V_d is defined as

$$V_d = \{u, r \big| u \in [u_a - \dot{u}_{min}T_a, u_a + \dot{u}_{max}T_a], \\ r \in [r_a - \dot{r}_{max}T_a, r_a + \dot{r}_{max}T_a]\} \quad (5)$$

where T_a is time interval during which the accelerations are constant. In addition, determined by the vessel's inherent properties, the space of possible velocities can be given by

$$V_s = \{u, r \big| u \in [u_{min}, u_{max}], r \in [-r_{max}, r_{max}]\} \quad (6)$$

Considering the three limitations above, the feasible area V_r is defined as

$$V_r = V_a \cap V_d \cap V_s \quad (7)$$

B. Selecting the Optimal Velocities

Through the Eq.7 the (u, r) can be calculated in a given time, however, the area in the collection is continuous resulting in inconvenient use. For this reason, the (u, r) is discretized in this algorithm. Then u is discretized into M velocities and r is discretized into N angular velocities. There will be $M \cdot N$ different choices in total.

When feasible discretization velocities are calculated, the next step is to choose the optimal velocities. In order to avoid obstacle optimally, three evaluation components are given different weights, then the maximum of the objective function is defined as

$$G(u, r) = \sigma(\alpha \cdot heading(u, r) + \beta \cdot dist(u, r) \\ + \gamma \cdot velocity(u, r)) \quad (8)$$

where

- $heading(u, r)$ is the component hopefully making the vessel move towards the target. It is defined as $180 - \theta$, where θ is the angle between a vessel actual heading and desired heading formed by the target point and vessel current point.

- $dist(u, r)$ is the distance to the adjacent obstacle that intersects with arc trajectories. If there is no obstacle around the vessel, the $dist(u, r)$ is a larger constant.
- $velocity(u, r)$ is the velocity of the vessel travels along the arc trajectory, in order to reach the goal faster, the larger velocity is selected.

In objective function, weight parameters α, β, γ are the constants ensure that the vessel reaches the goal on the optimal path at high velocity. The vessel moves freely with a low value α, in contrast, it moves closely to the obstacles with a high value α. Thus, higher values are set in narrow terrain and smaller values are more suitable in less obstacles terrain.

In marine practice, in order to avoid obstacles reliably, a safety margin around the obstacles is considered. It is common to expand a virtual area according to the shape of the obstacle. The size of this area is determined by the dynamics model of the vessel but must be at least more than twice the turning radius.

IV. Simulation Studies

Simulation experiments have been carried out both for global and local algorithm. Based on the way-point guidance law, the USV follow the global trajectory planned by A* PS. Then it could bypass unknown obstacles in local region using Dynamic Window. As the local obstacles are completely avoided, the USV continued to return to the global planning path. All simulations were performed on machine with Intel i7 3.4 GHz and 8GB RAM, running by the MATLAB R2016a.

A. Simulation of Global Path Planning

The global path planning algorithm is based on a real-world map simulation. For a real-world map, we should mesh it and label each cell as an obstacle or not. Thus, the first step is turning the color image into a gray image and then making it binarization. Assume that the gray image function $f(x, y)$ is input, then the output of binary image is given by

$$g(x, y) = \begin{cases} 0, & \text{if } f(x, y) < Thredshold \\ 255, & \text{if } f(x, y) > Thredshold \end{cases} \quad (9)$$

Based on the binarization result, the size of the color map was set to 1024 by 768 pixels grid map which cells are labeled as obstructed.

To verify the modified algorithm, A* and A* PS were compared in simulation. In Fig. 3, the green path is generated by A*, and the red is generated by A* PS. The obstacles have been expanded to 10×10 pixels for the purpose of making the path with a certain distance from the obstacles. According to TABLE I, the A* PS algorithm has shorter path length. It is clear that the number of way-points generated by A* PS is far less than traditional algorithm.

Fig. 3. Comparison of A* and A* PS.

B. Simulation of Local Collision Avoidance

In order to effectively avoid unknown obstacles, simulation experiments were conducted based on dynamic window method. Table II shows the main parameters used in this simulation. The dynamic window approach to collision avoidance in local region is depicted in Fig. 4. The USV started at the green dot, passed through the blue path and encountered unknown square obstacles. The green feasible trajectories are generated by discretization velocity space, and each trajectory represents a set of velocity and angular velocity. Finally, the red optimal trajectory is selected by maximizing the objective function $G(u, r)$ with incorporating the appropriate parameters α, β, γ.

Traditional dynamic window method does not take the safety margin around obstacles into consideration. Comparing Fig. 5(a) and Fig. 5(b), it is difference that the virtual area around the obstacles is expanded in the latter figure. It is obvious that collision danger will not happen in Fig. 5(b) because it is far away from the obstacle, which is more reliable to the former one.

In order to prove better performance of local collision avoidance, simulation experiments on a real scenario were conducted to select a set of optimal parameters α, β, γ. As shown in Figs. 6−8, a virtual safety zone pertaining to the shape of the obstacle has been established to ensure reliable navigation at high speed. In Fig. 6, the parameters of α, β, γ are 0.1, 0.8, 0.1, respectively, from which we can see that the USV sails away from obstacles because β accounts for a large proportion. In Fig. 7, the parameters of α, β, γ are set to 0.4, 0.4, 0.2, respectively, from which we can see that the USV is closer to obstacles than former figure and the time reaching destination is shorter than former one. Having considered the safety zone, the result in Fig. 8 shows better performance because of a high heading weight. Actually, in different environment, the best performance of obstacle avoidance can be achieved by selecting different parameters.

V. Conclusions

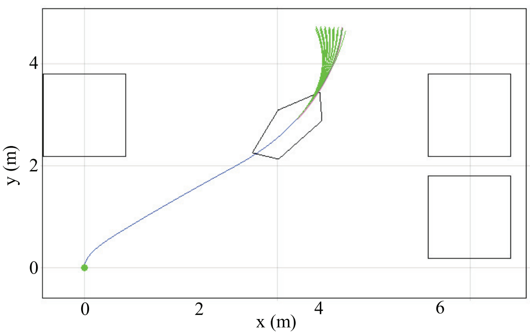
In this paper, a hybrid path-planning scheme combining global path planning with local collision avoidance has been proposed. Global way-point navigation using the A* PS al-

TABLE I
COMPARISON OF THE SIMULATION

Algorithm	Number of way-points	Path length	Runtime
A* PS	9	178.2 m	5.6784 s
A*	354	213.6 m	4.9987 s

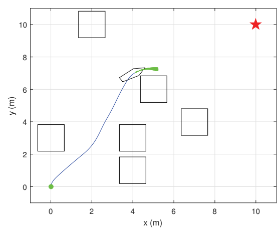
TABLE II
MAIN PARAMETERS IN DYNAMIC WINDOW

Maximum velocity	1.2 m/s
Maximum acceleration	0.2 m/s^2
Maximum angular velocity	0.35 rad/s
Maximum angular acceleration	0.87 rad/s^2
σ	1

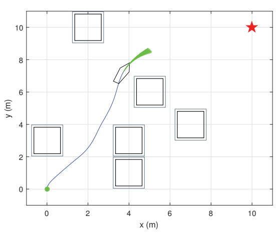


t

Fig. 4. The dynamic window approach.


t

(a) DW without a virtual safety zone



(b) DW with a virtual safety zone

Fig. 5. Comparison of dynamic window.

t

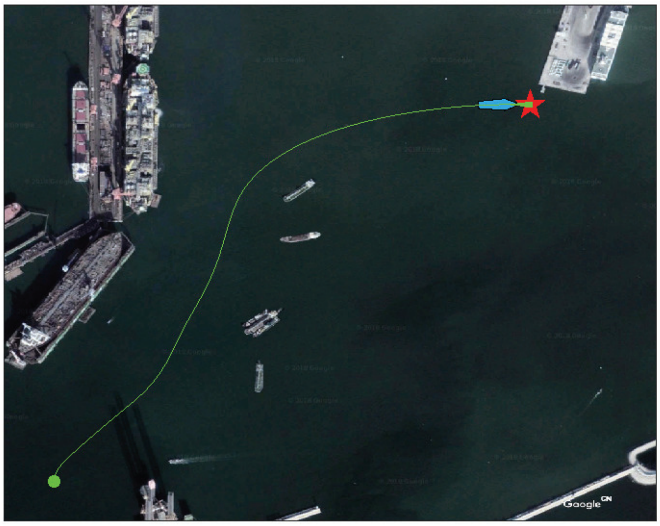


Fig. 6. DW1 ($\alpha = 0.1, \beta = 0.8, \gamma = 0.1$).

t

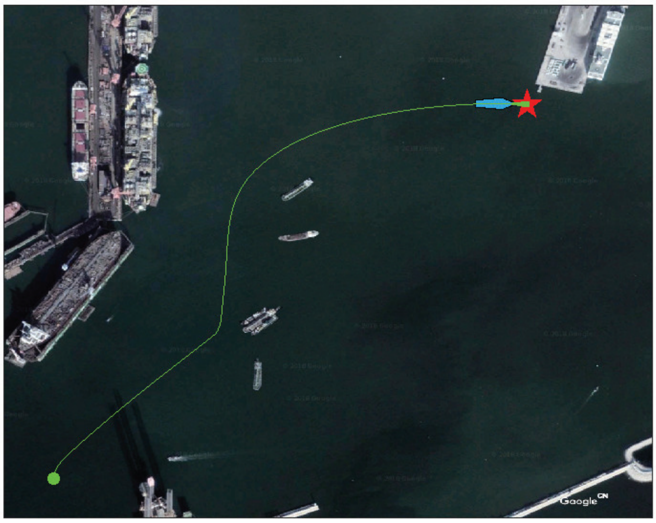


Fig. 7. DW2 ($\alpha = 0.4, \beta = 0.4, \gamma = 0.2$).

gorithm has been performed in a real-world map which can not only reduce the number of way-points but also generate a shorter path. By virtue of the Dynamic Window approach, the USV can avoid unknown obstacles via selecting optimal velocities. The virtual safety zone has been employed to ensure that the USV avoids obstacle reliably. Simulation results have demonstrated the effectiveness and superiority of the proposed hybrid path-planning scheme with collision avoidance.

ACKNOWLEDGMENT

The authors would like to thank anonymous referees for their invaluable comments and suggestions. This work is supported by the National Natural Science Foundation of P. R. China (under Grants 51009017 and 51379002), the Fund for Dalian Distinguished Young Scholars (under Grant 2016RJ10),

t

Fig. 8. DW3 ($\alpha = 0.7, \beta = 0.1, \gamma = 0.2$).

the Innovation Support Plan for Dalian High-level Talents (under Grant 2015R065), and the Fundamental Research Funds for the Central Universities (under Grant 3132016314).

REFERENCES

[1] N. Wang, S. L. Lv, W. D. Zhang, and Z. Z. Liu, "Finite-time observer based accurate tracking control of a marine vehicle with complex unknowns," *Ocean Engineering*, vol. 145, pp. 406-415, 2017.

[2] N. Wang, Z. Sun, and Z. J. Zheng, "Finite-time sideslip observer based adaptive fuzzy path following control of underactuated marine vehicles with time-varying large sideslip," *International Journal of Fuzzy Systems*, pp. 1-12, 2017.

[3] N. Wang, S. L. Lv, and Z. Z. Liu, "Global finite-time heading control of surface vehicles," *Neurocomputing*, vol. 175, pp. 662-666, 2016.

[4] N. Wang, S. L. Lv, M. J. Er, and W. H. Chen, "Fast and accurate trajectory tracking control of an autonomous surface vehicle with unmodelled dynamics and disturbances," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 3, pp. 230-243, 2017.

[5] N. Wang, J. C. Sun, M. J. Er, and Y. C. Liu, "A novel extreme learning control framework of unmanned surface vehicles," *IEEE Transactions on Cybernetics*, vol. 46, no. 5, 2016.

[6] S. Campbell, W. Naeem, and G. W. lrwin, "A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance maneuvers," *Annual Reviews in Control*, vol. 36, no. 2, pp. 267-283, 2012.

[7] D. Treglia, *Game Programming Gems*, Charles River Media Inc, 2000.

[8] R. R. Murphy, *Introduction to AI Robotics*, MIT Press, 2000.

[9] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 2, pp. 269-271, 1959.

[10] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100-107, 1968.

[11] N. S. Son and H. K. Yoon, "Study on a waypoint tracking algorithm for unmanned surface vehicle (USV)," *Journal of Navigation and Port Research*, vol. 33, no. 1, 2009.

[12] A. Stentz, "The focussed D* algorithm for real-time replanning," *International Joint Conference on Artificial Intelligence*, vol. 124, no. 4, 1995.

[13] T. L. Perez and M. A. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Communications of the Acm*, vol. 22, no. 10, 1979.

[14] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *IEEE International Conference on Robotics and Automation*, vol. 2, no. 5, pp. 500-505, 2003.

[15] Z. Pan, J. Q. Li, K. M. Hu, and H. Zhu, "Intelligent vehicle path planning based on improved artificial potential field method," *Applied Mechanics and Materials*, vol. 742, no. 1, 2015.

[16] I. Kamon, E. Rivlin, and E. Rimon, "A new range-sensor based globally convergent navigation algorithm for mobile robots," *IEEE International Conference on Robotics and Automation*, vol. 1, no. 1, pp. 429-435, 2002

[17] J. Ng and T. Unl, "Performance comparison of bug navigation algorithms," *Journal of Intelligent and Robotic Systems*, vol. 50, no. 1, pp. 73-84, 1995.

[18] I. Kamon, "Tangentbug: A range-sensor-based navigation algorithm," *International Journal of Robotics Research*, vol. 17, no. 9, pp. 934-953, 1998.

[19] I. Ulrich and J. Borenstein, "VFH+: reliable obstacle avoidance for fast mobile robots," *IEEE International Conference on Robotics and Automation*, vol. 2, no. 2, pp. 1572-1577, 2002.

[20] D. Fox, W. Burgand, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics and Automation Magazine*, vol. 4, no. 1, pp. 23-33, 2002.