INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research

International Journal of Vehicle Design

ISSN online: 1741-5314 - ISSN print: 0143-3369
https://www.inderscience.com/ijvd

Intelligent obstacle avoidance control method for autonomous vehicles based on improved SAC algorithm

Yulin Ma, Yide Qian, Teng Ma, Yicheng Li, Jian Wan

DOI: 10.1504/IJVD.2025.10073969

Article History:

Received:	25 December 2024
Last revised:	14 August 2025
Accepted:	15 August 2025
Published online:	04 February 2026

Intelligent obstacle avoidance control method for autonomous vehicles based on improved SAC algorithm

Yulin Ma, Yide Qian and Teng Ma

School of Mechanical and Automotive Engineering,
Anhui Polytechnic University,
Wuhu, 241000, China
Email: mayulin@mail.ahpu.edu.cn
Email: 2230111123@stu.ahpu.edu.cn
Email: 2230142102@stu.ahpu.edu.cn

Yicheng Li

Automotive Engineering Research Institute,
Jiangsu University,
Zhenjiang, 212013, China
Email: liyicheng070@ujs.edu.cn

Jian Wan*

School of Software Engineering,
Jinling Institute of Technology,
Jiangsu Key Laboratory of Digital Intelligent
Low-Carbon Transportation,
Nanjing, 211169, China
Email: wanjian@jit.edu.cn
*Corresponding author

Abstract: To improve the success rate of collision avoidance for autonomous vehicles and shorten response time, an intelligent obstacle avoidance control method based on an improved SAC algorithm is proposed. This method is based on a self-organising cluster model, integrating short-range repulsion, medium-range velocity calibration, and obstacle avoidance rules to achieve collision-free cluster collaboration. The conventional SAC algorithm adopts the AC framework to maximise the expected reward and entropy value while reducing the estimation bias of the value function through the value network component. On this basis, the PER-SAC method is proposed, which integrates priority experience replay (PER) and importance sampling weight strategy while optimising network structure, reward and punishment functions, continuous state and action space design. Additionally, transfer learning is incorporated. The experimental results demonstrate the effectiveness of this method, achieving a collision avoidance success rate of 97%, with a maximum response time of just 0.54 s.

Keywords: improved SAC algorithm; autonomous vehicles; intelligent obstacle avoidance control; PER; priority experience replay.

Reference to this paper should be made as follows: Ma, Y., Qian, Y., Ma, T., Li, Y. and Wan, J. (2025) 'Intelligent obstacle avoidance control method for autonomous vehicles based on improved SAC algorithm', *Int. J. Vehicle Design*, Vol. 99, No. 5, pp.1–19.

Biographical notes: Yulin Ma is currently serving as a Professor at the School of Mechanical and Automotive Engineering, Anhui Polytechnic University. He obtained his Doctor's degree in Intelligent Transportation Engineering from Wuhan University of Technology. His research focuses on automated driving planning and control, driving automation verification and validation.

Yide Qian received his BE from the Wanjiang University of Technology, China, in 2022. He is currently pursuing the ME with the School of Mechanical and Automotive Engineering, Anhui Polytechnic University, China. His research interests include trajectory planning and control, and automatic driving.

Teng Ma received his BE from Henan Institute of Technology, China, in 2022. He is currently pursuing the ME with the School of Mechanical and Automotive Engineering, Anhui Polytechnic University, China. His research interests include autonomous driving test evaluation.

Yicheng Li is an Associated Professor at the Automotive Engineering Research Institute, Jiangsu University, Zhenjiang, China. He received his PhD in Intelligent Transport Engineering from Wuhan University of Technology, China in 2018. His research interests include intelligent vehicles and intelligent transportation systems.

Jian Wan is a Professor-Level Senior Engineer at Jinling Institute of Technology, NanJing, China. He received his Master's degree in Intelligent Transportation from Wuhan University of Technology and his Doctor's degree in Electronics and Information from Southeast University. His research focuses on transportation artificial intelligence, including in-vehicle intelligent perception and control technologies and roadside equipment intelligent perception and analysis technologies.

1 Introduction

The rapid progress of science and technology has propelled autonomous vehicles, a key link in the intelligent transportation system, to gradual transition from academic discussion to practice (Huang et al., 2024). Autonomous vehicles integrate advanced sensor technology, artificial intelligence algorithms, high-precision maps, and powerful computing capabilities, aiming to achieve autonomous navigation, environmental perception, decision-making, and execution control of vehicles, thereby greatly improving road safety, traffic efficiency, and passenger comfort (Lv et al., 2021; Luo et al., 2024; Liu et al., 2023). However, in real-word road environments, autonomous vehicles face complex and ever-changing obstacle challenges, such as sudden pedestrian crossings, vehicle breakdowns, road construction areas, etc. which require rapid and accurate collision avoidance response to ensure vehicle safety (Gao et al., 2022). Conventional obstacle avoidance methods often rely on pre-set rules or simple sensor

data processing, struggle to adapt to highly dynamic and uncertain real-world road scenes. Therefore, developing an efficient and intelligent obstacle avoidance control method has become the key to advancing autonomous vehicle technology (De et al., 2022).

Bai et al. (2024) proposed a vehicle lane changing and obstacle avoidance control method based on laser point cloud. Through a specially designed laser point cloud data capture architecture, relevant data of the vehicle operating scene was obtained. By introducing advanced LiDAR image processing technology, the method achieved precise surrounding environmental perception for intelligent connected vehicles. By integrating obstacle recognition, lane recognition, and delineation of passable areas, this study deeply explored the interaction characteristics between intelligent connected vehicles and obstacles and lanes during lane changing manoeuvres. Through intelligent optimisation search strategies for laser point clouds, boundary points were gradually identified until the constructed path meets predefined standards. Based on the linear road segment model and the fundamental principle of lane continuity, automatic lane changing and lateral obstacle avoidance strategies for autonomous vehicles were planned and implemented. However, the process of generating laser point cloud data remains susceptible to environmental conditions, especially in extreme weather conditions (including rain, fog, snow) and strong lighting, which can lead to a decrease in data collection accuracy, thereby compromising subsequent environmental cognition, obstacle identification and other activities, and weakening the effectiveness of path changes and obstacle avoidance strategies. Lai et al. (2021) proposed a collision avoidance control method for unmanned vehicles based on DDPG algorithm, clarifying the input and output parameters of the control system, and designing a road turning trajectory planning scheme based on sine function to significantly improve the vehicle obstacle avoidance performance. A neural network controller was carefully designed for the given input-output parameters, with corresponding control strategies being thoroughly explored. In order to address the issue of scarce reward signals, a reward mechanism adjustment scheme based on logarithmic function characteristics was proposed and implemented to optimise the learning process and improve the system adaptability. The computational complexity of the DPG algorithm is considerable, which imposes strict requirements on hardware computing resources in real-world autonomous vehicle applications. Inadequate hardware performance may result in low algorithm execution efficiency, making it difficult to implement real-time collision prevention control strategies and adversely affecting the safety performance of unmanned vehicles. Chen et al. (2024) proposed an obstacle avoidance control method for unmanned vehicle formations based on graph and fluid disturbance algorithms. To overcome the challenge of controlling multiple unmanned vehicle clusters, the collaborative operation network of each unmanned workshop was depicted using graph theory tools. Based on communication range constraints and integrating backstepping theory, a cluster controller with a navigation-following mechanism was constructed. With the help of Lyapunov stability theory, it has been confirmed that the expected formation pattern exhibits asymptotic stability. We have designed an innovative path planning scheme based on fluid mechanics to overcome the challenge of obstacle avoidance in complex and changing obstacle environments. This plan designates the leading unmanned vehicle to calculate and guide the movement trajectory of the entire formation, thus ensuring collective obstacle avoidance safety. In the leader follower mode, the overall effectiveness of the fleet significantly depends on

the performance of the leading vehicle. Its navigation vehicle malfunctions (such as sensor failure) could disrupt the formation's path planning, compromising effective obstacle avoidance and jeopardising the overall stability of the formation.

In the intelligent obstacle avoidance control method for autonomous vehicles, the main challenges include precise perception in complex environments, decision planning in extreme scenarios, and real-time and stability of algorithms. The improved SAC algorithm integrates priority experience replay (PER) and importance sampling weight strategy. In addition, the introduction of transfer learning enables the algorithm to gradually train from simple to complex environments, further improving training speed and obstacle avoidance performance. These improvements enable the PER-SAC algorithm to achieve more stable and accurate intelligent obstacle avoidance control for autonomous vehicles. In address the existing problems in obstacle avoidance control methods for unmanned vehicles, this study proposes an intelligent obstacle avoidance control method for unmanned vehicles based on an improved SAC algorithm. The detailed technical route of this method is as follows:

1 Based on the micro individual interaction mechanism, short-range repulsion and medium-range velocity calibration rules are adopted, and wall avoidance and obstacle response rules, intelligent obstacle avoidance control are incorporated for autonomous vehicles. By constructing cluster behaviour rules and utilising a semi spring model to achieve short-range repulsion, it ensures no collisions between vehicles; additionally, an ideal braking curve is designed to achieve mid-range speed calibration, ensuring synchronised vehicle movement and collision prevention.

2 To optimise the intelligent obstacle avoidance control strategy for autonomous vehicles, this study improves the SAC algorithm and proposes the PER-SAC method. This method integrates PER and importance sampling weight strategy to improve the chances of selecting key samples and reduce model errors. By designing a reasonable network structure, continuous state and action space, as well as reward and punishment functions, the training speed and stability of the algorithm can be improved to ensure the safe driving of autonomous vehicles in complex environments.

3 By introducing transfer learning techniques, the inter-task correlations are explored to initialise parameters for related tasks through parameter transfer techniques, accelerating the learning process of autonomous vehicles' strategies in different contexts. Through model preloading and random initialisation training, parameters close to the target position and obstacle scene parameters are obtained, further refining obstacle avoidance rules and achieving intelligent obstacle avoidance control for autonomous vehicles.

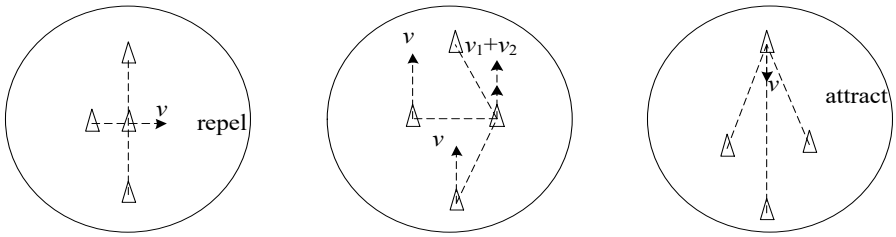
2 Intelligent obstacle avoidance control target for autonomous vehicles

Based on a design centred around microscopic individuals, achieving and maintaining collision-free group behaviour requires only three idealised mechanisms of interaction between individuals: short-range repulsion, medium-range velocity calibration, and long-range attraction (Zhao et al., 2024). Within this universal principle framework, numerous

models have been developed to explain collaborative cluster behaviour in animals, humans, and even cell migration. These systems are collectively referred to as self-organising systems. Recently, self-organising cluster models have gained new applications in robotic swarm science. As the primary prerequisite for ensuring safe operation, intelligent obstacle avoidance control strategies for autonomous vehicles have been explored based on these interactions (Tao and Du, 2022). This study adopts the first two of the three aforementioned interactions and additionally incorporate rules for wall avoidance and obstacle response.

The schematic diagram of cluster behaviour rules is shown in Figure 1.

Figure 1 Schematic diagram of cluster behaviour rules: (a) short-range repulsion; (b) mid-range speed calibration and (c) remote attraction



In Figure 1, it is observed that the process of constructing and maintaining collision-free clusters relies on the perception of the motion states of neighbouring individuals. At every decision-making moment, it is crucial to follow three basic principles: short-range exclusion, medium-range velocity consistency, and clustering towards the group centre.

1 *Short-range repulsion*

Regarding local repulsion, a basic semi spring model is used in this study, which centres on a central velocity component at a linear distance (Li et al., 2023). Within the set maximum operating range r_0^{rep}, the autonomous vehicle activates a mutual exclusion program, with the repulsion term expressed as:

$$r_{ij}^{rep} = \begin{cases} p^{rep} \cdot \left(r_0^{rep} - r_{ij}\right) \cdot \dfrac{r_i - r_j}{r_{ij}}, r_{ij} < r_0^{rep} \\ 0, r_{ij} \geq r_0^{rep} \end{cases} \tag{1}$$

In the formula, p^{rep} represents the repulsion coefficient, and r_{ij} represents the distance between two autonomous vehicles.

Compared to other vehicles, the total repulsion speed of autonomous vehicle i is:

$$v_i^{rep} = \sum_{i \neq j} v_{ij}^{rep} \tag{2}$$

2 *Mid-range speed calibration*

Given the interference of external or internal factors, such as lag in rejection reactions and noise-induced self-excited oscillations, a speed matching mechanism must be

implemented between autonomous vehicles to ensure synchronous movement in collective action (Deng et al., 2021). Additionally, a key prerequisite must be considered: if autonomous vehicles have limited acceleration capability, a longer following distance should be maintained at higher speed differentials to avoid collisions.

In order to achieve the aforementioned standards, the speed calibration benchmark is based on constructing an ideal braking curve. This curve achieves smooth attenuation of velocity in space, and its mathematical expression can be expressed as:

$$D(r,a,p)=\begin{cases}0, r\le 0\\ rp, 0<rp<a/p\\ \sqrt{2ar-a^2/p^2}, rp\ge a/p\end{cases} \tag{3}$$

In the formula, r is the distance between the unmanned vehicle and the expected stopping position; a represents the vehicle acceleration; and p represents the linear gain parameter (Lee et al., 2021).

The core principle of speed calibration is to ensure that the speed difference between two autonomous vehicles at a specific distance remains within the safe speed deviation range defined by the ideal braking trajectory. Its mathematical expression is:

$$v_{ij}^{frictmax}=max\left(v^{frict}, D\left(r_{ij}-r_0^{frict}, a^{frict}, p^{frict}\right)\right) \tag{4}$$

$$v_{ij}^{frict}=\begin{cases}C^{frict}\left(v_{ij}-v_{ij}^{frictmax}\right)\dfrac{v_i-v_j}{v_{ij}}, v_{ij}>v_{ij}^{frictmax}\\ 0, v_{ij}\le v_{ij}^{frictmax}\end{cases} \tag{5}$$

In the formula, C^{frict} represents the speed error coefficient; r_0^{frict} represents the distance between the expected stop of unmanned vehicle i and vehicle j; and v_{ij} represents the speed difference (Yan et al., 2021).

For the autonomous vehicle i, its overall speed adjustment formula relative to surrounding vehicles can be expressed as:

$$v_i^{frict}=\sum_{j\ne i}v_{ij}^{frict} \tag{6}$$

The functional diagram of the repulsion speed and distance between autonomous vehicles is shown in Figure 2.

3 *The interaction between obstacles*

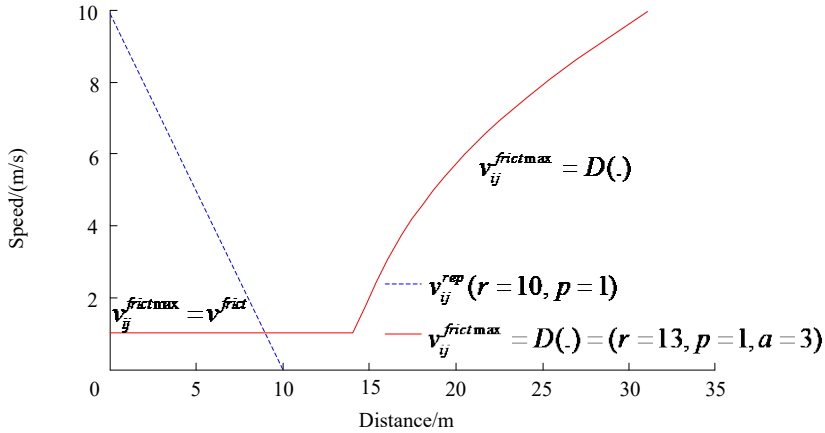
A virtual rectangular boundary with repulsive characteristics is constructed, which runs at a speed v^{shill} to maintain the coordinated movement of autonomous vehicles. Once vehicles approach these virtual boundaries, they should reduce their speed towards the boundaries. The expression for the virtual wall function is:

$$v_{is}^{shillmax}=D\left(r_{is}-r_0^{shill}, a^{shill}, p^{shill}\right) \tag{7}$$

$$v_{is}^{wall} = \begin{cases} \left(v_{is} - v_{is}^{shillmax}\right) \cdot \dfrac{v_i - v_s}{v_{is}}, v_{is} > v_{is}^{shillmax} \\ 0, v_{is} \le v_{is}^{shillmax} \end{cases} \tag{8}$$

In the formula, r_{is} represents the distance between the virtual wall and the unmanned vehicle, and v_{is} represents the speed difference between the virtual wall and the unmanned vehicle.

Figure 2 Schematic diagram of speed calibration (see online version for colours)



4 *Self-starting*

In addition to the mutual influence between vehicles and between vehicles and obstacles discussed previously, the expected speed of autonomous vehicle i also incorporates a basic self-driving component. This component aligns with the direction of the current velocity vector v_i and has a constant value v^{flock}.

5 *Final expected speed*

To obtain the desired velocity value, all the interaction force terms mentioned previously need to be vector summed:

$$\tilde{v}_i^d = \frac{v_i}{|v_i|} v^{flock} + v_i^{rep} + v_i^{frict} + \sum_s v_{is}^{wall} + \sum_s v_{is}^{obstacle} \tag{9}$$

After completing the superposition operation, a velocity limit v^{max} is added to maintain the direction of the desired velocity. If the desired speed exceeds this limit, its size should be adjusted to accommodate:

$$v_i^d = \frac{\tilde{v}_i^d}{|\tilde{v}_i^d|} min\left\{\left|\tilde{v}_i^d\right|, v^{max}\right\} \tag{10}$$

The v_i^d value calculated by formula (10) represents the instantaneous operating speed and direction of travel of the autonomous vehicle.

3 Improved SAC algorithm for intelligent obstacle avoidance control of autonomous vehicles

3.1 Traditional SAC algorithm

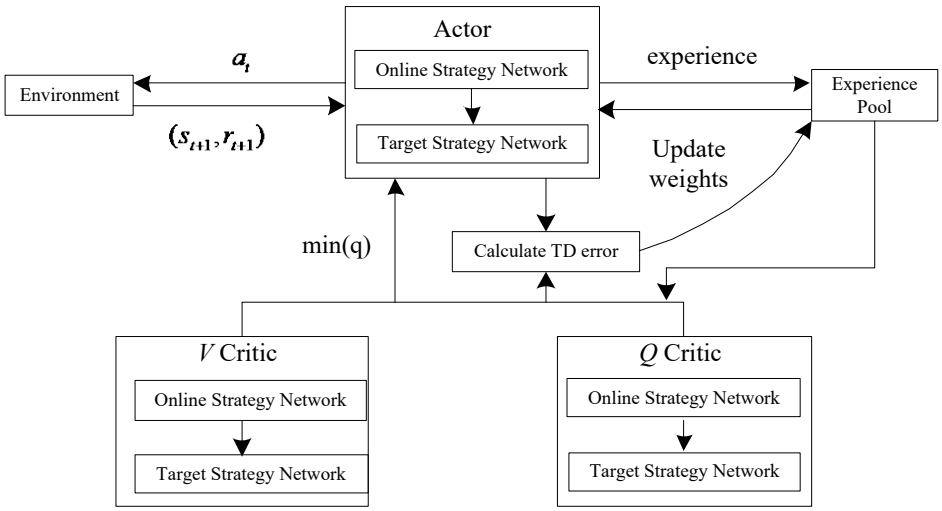
The SAC algorithm draws inspiration from the architecture of Actor Critic (AC). Traditional reinforcement learning methods focus on maximising the expected reward value, while SAC additionally pursues the maximisation of system entropy, striving to achieve both objectives:

$$J^*(\pi) = \underset{\pi}{\operatorname{argmax}} \sum_{t=0}^{T} E_{(s_t,a_t)\sim\rho_\pi}\left[r(s_t,a_t) + \tau H\left(\pi\left(a|s_t\right)\right)\right] \tag{11}$$

In the formula, the E value reflects the expected reward of the current situation, while the r value displays the immediate reward of the current situation. The set ρ_π encompasses all possible state action pairings; the H value measures the level of uncertainty (entropy) of the current action; the τ parameter regulates the impact of entropy; and the π ratio reveals the likelihood of taking action in the current situation.

The SAC algorithm introduces a value network component on top of the Actor Critic architecture to reduce bias in value function estimation. Its composition includes an Actor network and four Critic networks: the V and $TargetV$ networks are responsible for state value estimation, marked as $VCritic$; The Q_0 and Q_1 networks are responsible for estimating the action state value, denoted as $QCritic$. The network architecture of SAC algorithm is shown in Figure 3.

Figure 3 Network architecture of SAC algorithm



For a specific state s_t, the Actor network generates an action probability distribution map $\pi\left(a|s_t\right)$ and randomly selects an action $a_t \in a$ to execute based on this distribution. Next, action a_t is input into the environment, which returns a new state s_{t+1} and reward r_{t+1}, forming an experience: $\left(s_t, a_t, s_{t+1}, r_{t+1}\right)$, which is then stored in the experience pool.

In *QCritic* network training, the experience pool data $(s_t, a_t, s_{t+1}, r_{t+1})$ is extracted to update the network parameters ω, where the $q(s_t, a_t)$ value of action a_t is used as the estimated value of the s_t state, and the true value estimate of the s_t state is derived through the optimal Bellman equation:

$$U_t^{(q)} = r_t + \gamma E_\pi \left[R_{t+1} \middle| s_{t+1} = s \right] \tag{12}$$

In the formula, E_π is the cumulative expected return of the current state.

The square error loss function is used as the basis for training the *QCritic* network, and its mathematical expression is defined as follows:

$$Loss_Q = \frac{1}{|B|} \sum_{(s_t, a_t, s_{t+1}, r_{t+1}) \in B} \left[q(s_t, a_t; \omega) - U_t^{(q)} \right]^2 \tag{13}$$

In the formula, B is used to indicate the random selection of a data batch from an empirical database.

In the *VCritic* network, the data $(s_t, a_t, s_{t+1}, r_{t+1})$ collected by the experience pool is used to update the network parameters θ, while the *VCritic* network produces the following true estimates:

$$U_t^{(v)} = \sum_{a_t' \in A(s_t)} \pi(a_t' | s_t, \theta) \left[minq(s_t, a_t; \omega) - \alpha ln\pi(a_t' | s_t, \theta) \right] \tag{14}$$

In the formula, a_t' represents the subsequent set of possible actions estimated by the Actor network based on its policy parameter π, while $ln\pi(a_t' | s_t, \theta)$ measures the entropy of these actions.

The loss of *VCritic* network is calculated based on the true value:

$$Loss_V = \frac{1}{|B|} \sum_{(s_t, a_t, s_{t+1}, r_{t+1}) \in B} \left[v(s_t; \omega) - U_t^{(v)} \right]^2 \tag{15}$$

In the training process of Actor network, a loss function is defined, which is optimised through gradient descent method:

$$Loss_A = \frac{1}{|B|} \sum_{(s_t, a_t, s_{t+1}, r_{t+1}) \in B} E_{a_t \sim \rho_\pi} \left[q(s_t, a_t') - \alpha ln\pi(a_t' | s_t, \theta) \right]^2 \tag{16}$$

In reinforcement learning uses, the temporal difference (TD) error serves as the basis for adjusting the update amplitude, and the effect of the slight action a_t is evaluated by calculating the TD error:

$$\delta_t = r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \tag{17}$$

In the formula, Q represents the state value of Critic evaluation, while γ represents the discount rate of future rewards.

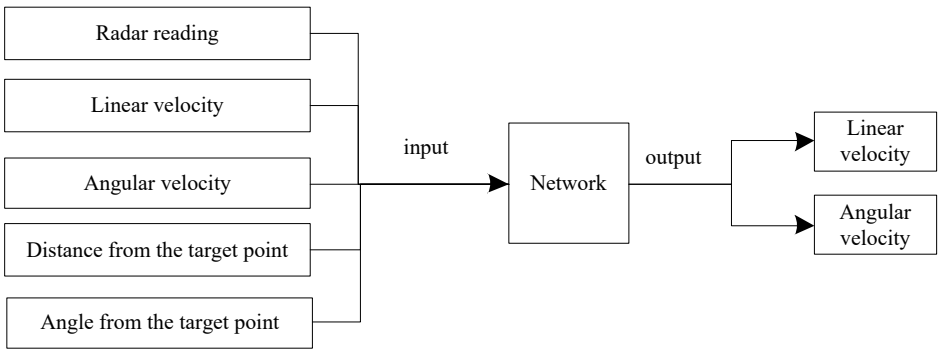
3.2 Improve SAC algorithm

This study proposes a novel PER-SAC strategy to accelerate training efficiency and ensure smoothness in the optimisation process. This strategy creatively embeds the principle of PER into the SAC algorithm, achieving a transition from obtaining samples from the experience pool based on sample priority rather than pure randomness. This modification greatly increases the selection probability of important samples, significantly improving learning performance. To address the bias introduced by priority replay, importance sampling weighting technique was introduced, and the loss function was accordingly revised to more effectively reduce model prediction error. The PER-SAC algorithm integrates network architecture design, reward system construction, and planning for continuous states and action domains.

1 *Network structure*

The neural architecture relied upon by the PER-SAC algorithm consists of 14 inputs and 2 outputs, as shown in Figure 4.

Figure 4 Input and output of the network



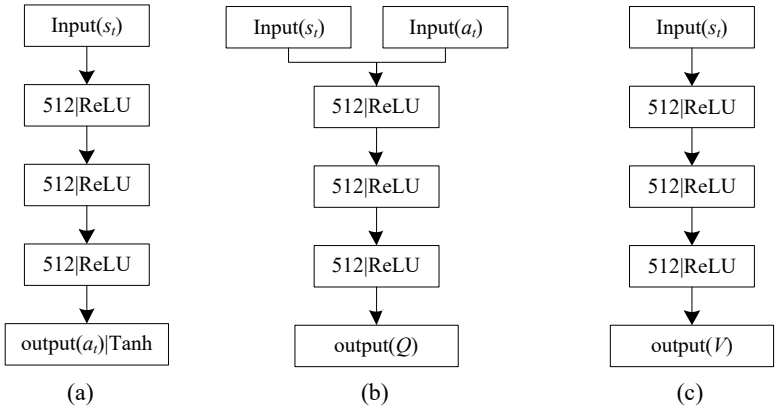
In Figure 4, the input data received by the network includes the measurement values x_i of the radar in 10 directions, the linear velocity v_{t-1} and angular velocity w_{t-1} of the autonomous vehicle, as well as the straight-line distance d_t and directional angle θ_t of the vehicle relative to the target. The output of the network is the linear velocity v_t and angular velocity w_t of the autonomous vehicle.

The SAC architecture consists of three core components: the policy network (Actor), the Q network (Q Critic), and the value evaluation value network (V Critic). The schematic diagram of the structure is shown in Figure 5.

2 *Replay based on priority experience*

The priority allocation mechanism in experience replay sets different importance levels for each sample. During sampling, the probability of selecting high-priority samples is increased according to their assigned importance levels, thereby accelerating training. The priority values of these samples are stored and managed using a SumTree data structure.

Figure 5 Schematic diagram of SAC network structure: (a) strategic network; (b) Q network and (c) V network



The sample priority is determined based on the magnitude of the time difference (TD) error, where a larger TD error corresponds to a higher sample priority. Formula (17) can be referred for the specific calculation method of TD error. The probability of sample extraction is obtained according to the following formula:

$$P(i)=\frac{p_i^a}{\sum_{i=1}^{k} p_i^\alpha} \tag{18}$$

In the formula, a represents the priority adjustment parameter, and p_i represents the priority of the sample.

When calculating TD error, three networks (Q network, value network and policy network) in SAC algorithm should be taken into account. Since the output values of Q network and value network are significantly greater than those of policy network, direct accumulation of their respective errors would correspondingly weak the contribution of the policy network' errors to the overall error. In order to achieve the above purpose, the adjustment factors T_α and T_β are used to implement appropriate correction on the output values of Q network and value network:

$$\delta_i=\left|T_\alpha \cdot TD(Q)\right|+\left|T_\beta \cdot TD(V)\right|+\left|TD(\pi)\right| \tag{19}$$

Priority experience replay has changed the sample extraction mode, so importance sampling weights are used to correct the errors introduced by this mode, while calculating the loss function in gradient training to reduce model errors. The mathematical expression for the importance sampling weight is as follows:

$$ISWeight=w_j=\frac{\left(N\cdot p_j\right)^{-\beta}}{max_i\left(w_i\right)} \tag{20}$$

In the formula, the weight of sample j is represented by w_j; $max_i\left(w_i\right)$ represents the sample with the highest weight; N represents the total number of sample sets; and β is the coefficient for weight adjustment.

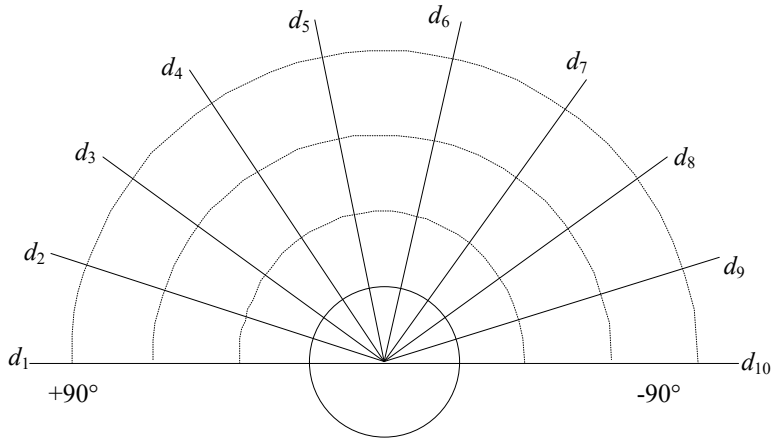
In the final stage, by applying importance sampling weights, the loss functions of both the Q network and the value network were updated accordingly. The update rules are detailed in formulas (13) and (15):

$$\begin{cases} Loss_Q^* = Loss_Q \cdot w_j \\ Loss_V^* = Loss_V \cdot w_j \end{cases} \tag{21}$$

3 *Continuous mobile space and state space design*

A reasonable continuous state domain and action space are designed as the input and output interface of the neural network. With sensors, detailed information about the surrounding environment is captured. This state domain reflects the current environmental conditions of autonomous vehicles and is the basis of their decision-making action space. The laser radar equipped on autonomous vehicles has a 360 ° scanning range and a maximum detection distance of 35 meters. Considering the lack of vehicles' reverse operation, the large amount of radar data, and the high computational requirements, only the detection results within a 180 ° field of view in front of the vehicle are used, and 10 specific directions of radar data are selected for processing. The data acquisition structure of LiDAR is shown in Figure 6.

Figure 6 Data acquisition structure of LiDAR



For autonomous vehicles, their motion status information integrates the nearest obstacle distance data d_i from 10 directional radars, as well as the straight-line distance D_g and angular deviation θ_g from the current position of the vehicle to the target location. Based on this information, the state space of autonomous vehicles can be defined as follows:

$$s_j = \left(d_1 \sim d_{10}, D_g, \theta_g\right) \tag{22}$$

4 *Design of reward and punishment function*

The design of the reward and punishment mechanism is a standard for measuring the quality of the next driving action selected by autonomous vehicles in a specific state. In order to cope with the scarcity of reward signals, a continuous reward and punishment

function design scheme is adopted. The mathematical expression of the reward and punishment function is as follows:

$$r(s_t, a_t) = \begin{cases} r_{arrive}, d_t < c_d \\ r_{collide}, min_x < c_o \\ c_{r1}(d_{t-1} - d_t), (d_{t-1} - d_t) > 0 \\ c_{r2}, (d_{t-1} - d_t) \le 0 \end{cases} \tag{23}$$

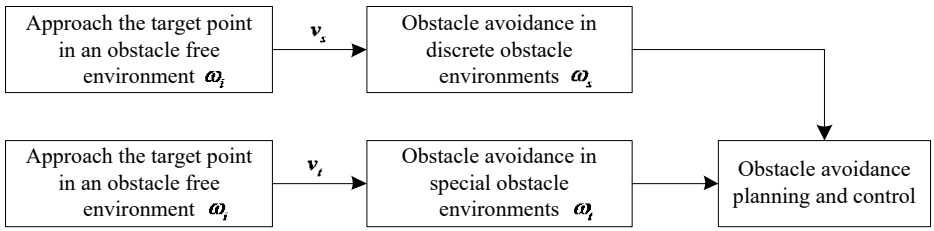
In the formula, r_{arrive} represents the forward establishment of reaching the target point; d_t represents the distance between the unmanned vehicle and the target position at the current time; and c_d represents the distance threshold. When encountering obstacles, $r_{collide}$ represents the negative reward received; min_x is the minimum distance value measured by the LiDAR; c_o represents the safe distance required to avoid collision; and c_{r1} and c_{r2} are the two parameter factors in reward calculation.

5 *Transfer learning*

When autonomous vehicle implement intelligent obstacle avoidance control, correlations exist among multiple tasks. By applying parameter transfer techniques in different environments and setting initial parameters for related tasks, the learning process of autonomous vehicles' strategies in different contexts can be effectively accelerated.

In the model preloading stage, all parameter sets of the model are firstly acquired. By utilising the random initialisation method for training, the parameter ω_s close to the target position and the parameter ω_t of the obstacle scene are obtained, further refining the obstacle avoidance rules v_s and v_t to realise intelligent obstacle avoidance control. The transfer learning framework is shown in Figure 7.

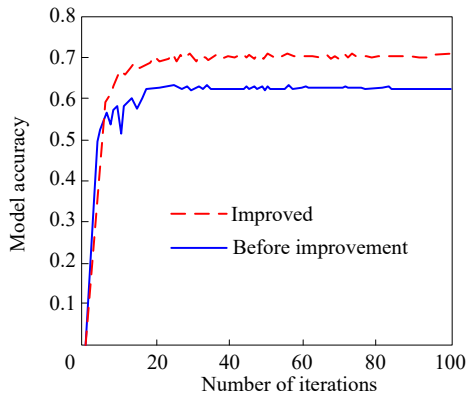
Figure 7 Transfer learning framework



The convergence curves of the SAC algorithm before and after improvement are shown in Figure 8.

Figure 8 clearly shows that compared to the conventional SAC algorithm, the improved SAC algorithm has delivered significantly improved accuracy and more precise pre-decision results. At the same time, the convergence curve of the algorithm also shows greater stability, with significantly reduced fluctuation amplitude, demonstrating its superior performance and stronger robustness.

Figure 8 Convergence curves of SAC algorithm before and after improvement (see online version for colours)



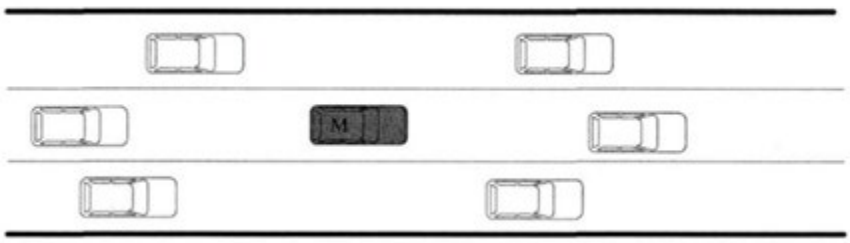
4 Test experiment

4.1 Experimental environment

The initial simulation parameters of the autonomous vehicle were set as follows: a 3.75 m central lane width with 3.5 m adjacent lane spacing, LiDAR configured for 10 Hz scanning frequency, with ±2 cm ranging accuracy, 270° horizontal field of view angle, and 120 maximum detection distance; wheel speed sensor operating at 100 Hz sampling frequency with ±0.1 m/s measurement error of linear velocity; IMU featuring ±300°/s angular velocity range; high-precision map with <10 cm positioning error, and 5 Hz target point coordinate update frequency. The initial speed was set to 60 km/h, with 4 m/s^2 maximum braking deceleration and 15 m minimum safe following distance.

During operation, an autonomous vehicle will encounter obstacles from the left front, left rear, right front, right rear, front and rear directions. In order to pursue the rigour of experimental results, a test scenario was designed in a three lane environment where obstacle vehicles appeared simultaneously in all directions. The schematic diagram of three lanes is shown in Figure 9.

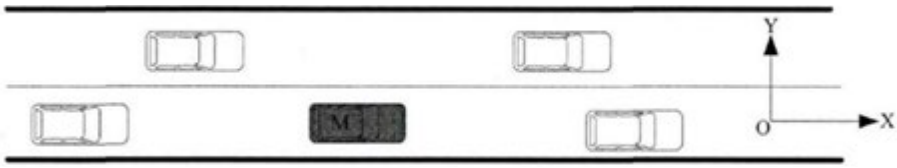
Figure 9 Schematic diagram of three lanes



The autonomous vehicle demonstrates symmetrical upward and downward lane-changing behaviour for obstacle avoidance when positioned in the central lane. At this time, the model can be reduced to a two lane obstacle avoidance model. If autonomous vehicles

need to change lanes from the upper or lower lane, they must all face the central lane for lane change. At this point, the model can also be simplified as a two lane change to avoid obstacles. The two lane changing model is shown in Figure 10.

Figure 10 Schematic diagram of two lane changing



In the obstacle avoidance control process of autonomous vehicles, the collection of LiDAR data is achieved through the LiDAR installed on the vehicle. LiDAR continuously emits laser signals, which are reflected back when encountering surrounding objects. Based on the round-trip time of the signals, the distance to the object can be calculated. Through continuous scanning, a point cloud data can be accumulated, which contains multiple types of environmental information and is the key to identifying obstacles. The accurate acquisition of linear velocity depends on the wheel speed sensors equipped on the wheels. When the wheels rotate, the sensors immediately start, accurately recording the wheel speed. Combined with key parameters such as wheel size, the vehicle's straight-line travel speed can be calculated through professional algorithms. This speed indicator intuitively reflects the speed characteristics of the vehicle's straight-line travel. As for angular velocity, it needs to be measured by an inertial measurement unit (IMU). The IMU is installed on the car and can sense the rotation of the car, accurately measuring angular velocity, which is very helpful for understanding the steering of the car. Target point location collection utilises pre-loaded high-precision map information such as target location, combined with the ability of cameras to recognise road signs, environmental features, and laser radar to detect object positions. Then, through the vehicle positioning system, the position of the target point relative to the vehicle is comprehensively determined, providing navigation basis for obstacle avoidance control.

4.2 Experimental scheme

Using obstacle avoidance success rate and obstacle avoidance response time as indicators, this method was compared and tested with the methods in Lai et al. (2021) and Chen et al. (2024).

Collision avoidance success rate: Collision avoidance success rate is a key indicator for measuring the effectiveness of intelligent obstacle avoidance control methods for autonomous vehicles. It refers to the probability of autonomous vehicles in successfully avoiding collisions with obstacles during driving, reflecting the effectiveness of intelligent obstacle avoidance control methods in preventing vehicle collisions.

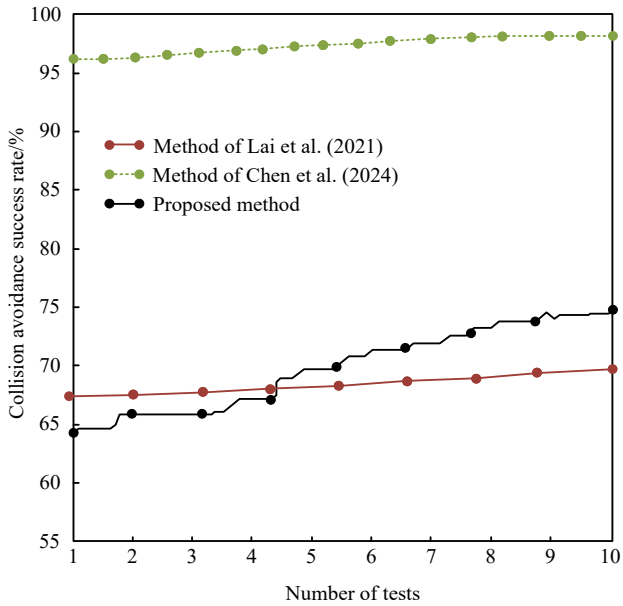
Obstacle avoidance response time: Obstacle avoidance response time refers to the time interval between an unmanned vehicle's sensor detection of an obstacle and its initiation of avoidance manoeuvres. A shorter obstacle avoidance response time enables vehicles to respond more quickly to potential collision hazards, thereby improving driving safety.

4.3 Experimental results

1 *Collision avoidance success rate*

The success rate of collision avoidance is the core indicator for measuring the effectiveness of intelligent obstacle avoidance control methods. In actual traffic scenarios, there are various types and complex obstacles, such as sudden pedestrian crossings and emergency braking of vehicles. The success rate of collision avoidance intuitively reflects the ability of autonomous vehicles to cope with these potential collision hazards. The test results of collision avoidance success rates for the three methods are shown in Figure 11.

Figure 11 Collision avoidance success rate (see online version for colours)



From the comparison of collision avoidance success rates shown in Figure 11, it can be clearly observed that the autonomous vehicle controlled by the method proposed in this paper exhibits excellent performance in collision avoidance. Specifically, the highest collision avoidance success rate of this method reached 97%, which fully demonstrates its high reliability and stability in complex road environments. At the same time, although the methods proposed in Lai et al. (2021) and Chen et al. (2024) also show a continuous upward trend in collision avoidance success rate, their highest collision avoidance success rate is only maintained at around 75%. This comparative result not only highlights the significant advantages of our method in collision avoidance performance, but also further verifies the effectiveness and reliability of this method in improving the safety of autonomous vehicles. Further analysis reveals that the excellent collision avoidance success rate achieved by the method proposed in this paper is mainly due to its innovative algorithm design and real-time data processing capabilities. This method adopts multi-sensor fusion technology, which can obtain accurate information about the surrounding environment of the vehicle in real time and make quick decisions

through deep learning models. In contrast, although the methods in Lai et al. (2021) and Chen et al. (2024) are progressiveness in theory, they are limited by computational efficiency and environmental adaptability in practical applications, resulting in a relatively low success rate of collision avoidance.

2 *Obstacle avoidance response time*

It is necessary to conduct obstacle avoidance response time testing for autonomous vehicles. It is directly related to the safety of vehicles, and a shorter obstacle avoidance response time allows vehicles to avoid obstacles in a timely manner, reduce collision risks, and ensure the safety of passengers, pedestrians, and other road users. This test helps evaluate the performance of the intelligent obstacle avoidance control system, and provides a quick and accurate response that reflects the system's efficiency. If the response time is too long, it may indicate problems in the perception, decision-making, or execution stages of the system. The test results of obstacle avoidance response time for the three methods are shown in Table 1.

Table 1 Obstacle avoidance response time

	Obstacle avoidance response time/s		
Number of tests	*Proposed method*	*Lai et al. (2021) method*	*Chen et al. (2024) method*
1	0.45	3.38	2.29
2	0.52	3.41	2.31
3	0.49	3.39	2.28
4	0.51	3.43	2.30
5	0.47	3.37	2.27
6	0.53	3.42	2.32
7	0.46	3.36	2.26
8	0.50	3.40	2.29
9	0.54	3.44	2.33
10	0.48	3.38	2.28

From the obstacle avoidance response time data in Table 1, it can be clearly seen that this method has significant advantages. Across 10 repeated tests, the obstacle avoidance response time of the proposed method is always kept within the optimal range of 0.45–0.54 s, exhibiting both stability and reliability. This represents an order of magnitude improvement compared with the existing methods. Specifically, the response time of the method in Lai et al. (2021) is up to 3.36–3.44 s. Although the method in Chen et al. (2024) is improved, it still takes 2.26–2.33 s to complete the obstacle avoidance response. This significant performance gap fully proves the innovative breakthrough of this method in algorithm design and system implementation. From the perspective of technical implementation, the superior speed of this method mainly attributes to the optimisation of the decision algorithm, which simplifies the traditional multi-level decision-making process into an end-to-end rapid response mechanism. In contrast, the reference method still uses the traditional serial processing architecture, which has obvious delay accumulation in data acquisition, feature extraction and decision execution.

5 Conclusion

This study has effectively improved the intelligent obstacle avoidance control method for autonomous vehicles by introducing an improved SAC algorithm, namely the PER-SAC method. Compared to traditional SAC algorithms, the PER-SAC method exhibits significant advantages in both collision avoidance success rate and obstacle avoidance response time. This method optimises the algorithm structure to reduce value function estimation bias, while improving the learning efficiency and performance of the algorithm through innovative means such as PER mechanism and importance sampling weight strategy. Experimental verification shows that this method has excellent obstacle avoidance ability and high safety in practical applications, providing new solutions and ideas for intelligent obstacle avoidance control of unmanned vehicles.

Acknowledgements

This work was supported in part by the Natural Science Research Project of Anhui Educational Committee under Grant No. 2023AH020015; and in part by the Applied Basic Research Project of Wuhu under Grant No. 2023jc10.

Conflicts of interest

All authors declare that they have no conflicts of interest.

References

Bai, X., Zhao, Y. and Wen, G.Q. (2024) 'Intelligent connected vehicle autonomous lane changing and lateral obstacle avoidance using laser point cloud', *Laser Journal*, Vol. 45, No. 6, pp.238–242.

Chen, Q., Pang, W. and Zhu, D.Q. (2024) 'Research on multi-unmanned ground vehicle formation and obstacle avoidance based on graph and fluid disturbance algorithm', *Control Engineering of China*, Vol. 31, No. 4, pp.669–677.

De, M.A., Iavernaro, F. and Mazzia, F. (2022) 'A minimum-time obstacle-avoidance path planning algorithm for unmanned aerial vehicles', *Numerical Algorithms*, Vol. 89, No. 4, pp.1639–1661.

Deng, W., Guo, H. and Zhang, K. (2021) 'Obstacle-avoidance algorithm design for autonomous vehicles considering driver subjective feelings', *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, Vol. 235, No. 4, pp.945–960.

Gao, D., Zhou, P. and Shi, W. (2022) 'A dynamic obstacle avoidance method for unmanned surface vehicle under the international regulations for preventing collisions at sea', *Journal of Marine Science and Engineering*, Vol. 10, No. 7, pp.252–258.

Huang, H., Ma, W.H. and Li, J.C. (2024) 'Intelligent obstacle avoidance control method for unmanned aerial vehicle formations in unknown environments', *Journal of Tsinghua University (Science and Technology)*, Vol. 64, No. 2, pp.358–369.

Lai, J.P., Li, H. and Shi, Y. (2021) 'Anti collision control strategy of unmanned vehicle based on DDPG algorithm', *Journal of Wuhan University of Technology*, Vol. 43, No. 10, pp.68–76.

Lee, H.Y., Ho, H.W. and Zhou, Y. (2021) 'Deep learning-based monocular obstacle avoidance for unmanned aerial vehicle navigation in tree plantations: faster region-based convolutional neural network approach', *Journal of Intelligent and Robotic Systems: Theory and Application*, Vol. 101, No. 1, pp.428–436.

Li, J., Chavez-Galaviz, J. and Azizzadenesheli, K.M.N. (2023) 'Dynamic obstacle avoidance for USVs using cross-domain deep reinforcement learning and neural network model predictive controller', *Sensors*, Vol. 23, No. 7, pp.349–362.

Liu, G., Wen, N. and Long, F. (2023) 'A formation control and obstacle avoidance method for multiple unmanned surface vehicles', *Journal of Marine Science and Engineering*, Vol. 11, No. 12, pp.141–149.

Luo, Q., Wang, H. and Li, N.Z.W. (2024) 'Multi-unmanned surface vehicle model-free sliding mode predictive adaptive formation control and obstacle avoidance in complex marine environment via model-free extended state observer', *Ocean Engineering*, Vol. 293, No. 1, pp.1–17.

Lv, J., Qu, C. and Du, S. (2021) 'Research on obstacle avoidance algorithm for unmanned ground vehicle based on multi-sensor information fusion', *Mathematical Biosciences and Engineering*, Vol. 18, No. 2, pp.1022–1039.

Tao, Y. and Du, J. (2022) 'Agile collision avoidance for unmanned surface vehicles based on collision shielded model prediction control algorithm', *Journal of Navigation*, Vol. 75, No. 5, pp.1243–1267.

Yan, X., Jiang, D. and Miao, R. (2021) 'Formation control and obstacle avoidance algorithm of a multi-USV system based on virtual structure and artificial potential field', *Journal of Marine Science and Engineering*, Vol. 9, No. 2, pp.115–126.

Zhao, J., Yu, H. and Fei, H.L. (2024) 'Path planning of unmanned vehicles based on adaptive particle swarm optimization algorithm', *Computer Communications*, Vol. 216, No. 25, pp.112–129.