Ocean Engineering 329 (2025) 121120

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

Research paper

A moving obstacle avoidance strategy-based collaborative formation for the bionic multi-underwater spherical robot control system

Ao Li [a], Shuxiang Guo [b,a,*], Chunying Li [b,**], He Yin [a], Meng Liu [a], Liwei Shi [a]

[a] Beijing Institute of Technology, Beijing, 100081, China
[b] Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen, Guangdong, 518055, China

ARTICLE INFO

Keywords:
Collaborative formation control strategy
Multiple moving obstacle avoidance
Multi-underwater spherical robots
Trajectory tracking
Adaptive model predictive control
Velocity obstacle (VO)-Based dynamic constraints

ABSTRACT

The collaborative operation of multi-underwater robot formation is an effective way to deal with the complex underwater environment. A formation control strategy with high efficiency and accuracy in moving obstacle avoidance is very important. Based on the multi-robot experiment platform, a moving obstacle avoidance-constrained adaptive model predictive control (MOAC-AMPC) strategy is proposed for Underwater Spherical Robots (USRs). The strategy is optimized in two main aspects: Firstly, based on model predictive control, an adaptive weight matrix based on tracking error is designed to solve the tedious parameter tuning problem and reduce the tracking time. Then, the Velocity Obstacle (VO)-based dynamic constraints are designed to avoid multiple moving obstacles. Finally, the multi-underwater spherical robots experiment platform is built. Pool experiments are set up on the platform to verify the multi-robot formation with obstacles avoidance. The feasibility and superiority of the proposed strategy are verified by simulations and experiments. The application of the proposed multi-USRs strategy has certain practical value in multi-robot trajectory tracking and obstacle avoidance.

1. Introduction

In recent years, the research of multiple underwater robots has attracted more attention (Yin et al., 2022; Han et al., 2022; Liu et al., 2021). The robot formation composed of multiple underwater robots has robustness and may effectively perform large-scale search and exploration tasks, such as underwater rescue, resource exploration, underwater equipment repair and so on (Peng et al., 2024; Li and Guo, 2023; Yin et al., 2023). The control methods applied to formation are the core part of multi-robot systems (Fu et al., 2020; Pan et al., 2022). Leader-follower method is widely used in formation and its control (Heshmati-Alamdari et al., 2021). The leader follows its desired trajectory. The follower maintains a predefined geometric relationship with the leader. However, failure of the leader may lead to failure of formation maintenance. The tracking errors of the followers accumulate with the increase of the tracking distances. The behavior-based method (Wen et al., 2023) uses weights to combine various patterns of formation behavior such as forming formation, achieving rendezvous, avoiding obstacles, etc. As the permission of some conflicting behaviors, the behavior-based method cannot ensure convergence with theoretical analysis. The virtual structure method uses a reference point as a leader. The desired trajectory points of all formation members are determined by the reference point (Zhen et al., 2022). However, the real-time virtual structure method of path planning is not flexible to the collision avoidance of moving obstacles. When one robot encounters an obstacle, the path of the other robots is affected. Many studies have further promoted the improvement of formation performance. Zhao et al. proposed a switching dynamic event-triggered prescribed performance formation control algorithm to satisfy the high-precision formation control task (Zhao et al., 2024). However, the balance between flexibility and reliability still needs to be further explored in the research of underwater robot formation.

The formation trajectory tracking is a basic problem of multi-underwater robot systems. The formation trajectory tracking needs to consider time to ensure that each member reaches the local target point at the corresponding time. Algorithms commonly used to control underwater robots for trajectory tracking include Active Disturbance Rejection Control (ADRC) (Tang et al., 2021), Proportional Integral Derivative (PID) Control (Shi et al., 2020) and backstepping (Peng et al.,

* Corresponding author. Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen, Guangdong, 518055, China.
** Corresponding author.
E-mail addresses: liao@bit.edu.cn (A. Li), guo.shuxiang@sustech.edu.cn (S. Guo), licy@sustech.edu.cn (C. Li).

https://doi.org/10.1016/j.oceaneng.2025.121120
Received 22 March 2024; Received in revised form 25 March 2025; Accepted 30 March 2025

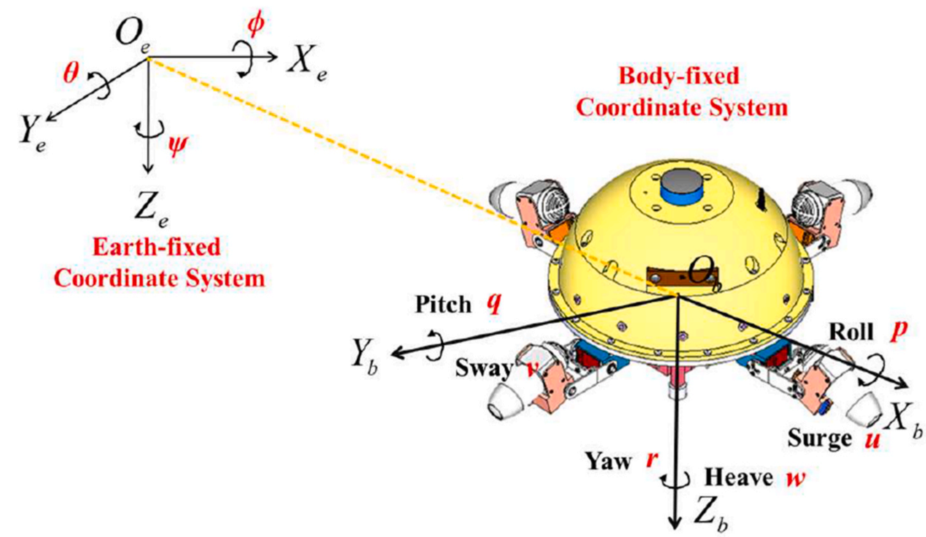

Fig. 1. The earth-robot coordinate system of USR.

2022). Constraints of the actual environment in robot control need to be considered, such as actuator saturation or the safe distance to obstacles. Considering these constraint conditions, the above control algorithms may not achieve the optimal control performance. In contrast, Model Predictive Control (MPC) simultaneously handles system constraints and achieves optimal control performance (Li et al., 2022a). Many researchers have used MPC to solve the formation trajectory tracking problem (Wei et al., 2021; Erskine et al., 2021). Wang et al. designed an adaptive MPC method based on Laguerre function. The recursive least squares algorithm was introduced to identify the model parameters of the system, to realize adaptive MPC (Wang et al., 2022). Wang et al. proposed an MPC method based on the lateral motion compensator and obtained the control output (Wang et al., 2023). However, many MPC-based algorithms did not have an off-line adjustment of weight matrix or real-time parameter identification, which limits their application in time-varying environments (Hou et al., 2024). Therefore, we expect to adjust the parameters of MPC adaptively to optimize the control process.

In the multi-underwater robot systems, obstacle avoidances must be considered to ensure the safety of formation members, that is, to avoid collision with other formation members and obstacles. Compared with static obstacles, obstacle avoidances for moving obstacles with randomness and uncertainty is more challenging and practical (Li et al., 2022b; Li and Guo, 2022). Pang et al. designed a variable formation reconstruction and obstacle avoidance control scheme based on affine transformation and improved artificial potential field (Pang et al., 2024). Meng et al. realized obstacle avoidances among underwater vehicles by adding a carefully designed artificial potential field cost term into the formation tracking cost function (Meng et al., 2023). In the above method, the relative speed was not taken into account, which cannot effectively avoid the obstacles (Hou et al., 2024; An et al., 2022a). The Velocity Obstacle (VO) method is a local obstacle avoidance algorithm that considers both relative position and relative velocity (An et al., 2022b). Therefore, based on the velocity cone of the VO method between the robot and the obstacle, the safe velocity is calculated on the convex polygon. This property can be used for convex optimization problems of MPC.

Based on affine transform and the improved artificial potential field, Pang et al. proposed a formation reconfiguration scheme with obstacle avoidance (Pang et al., 2024). The method adjusts the formation according to the obstacles in real time, and makes tracking and obstacle avoidance actions. Zhen et al. developed an advanced formation control and obstacle avoidance strategy for autonomous underwater vehicles (AUVs) in SE(3) based on gyroscopic force, and verified this through simulations (Zhen et al., 2024). Ding et al. proposed an AUV formation obstacle avoidance method based on virtual structure and artificial potential field method which are verified by simulation (Ding et al., 2024). Although the above methods achieve formation obstacle avoidance, there are some aspects that need to be improved, such as the formation needs to be re-planned due to obstacles, the obstacle avoidance is not flexible, or the lack of relevant experiments and considering realistic constraints.

To perform the task of exploration in the shallow marine environment with moving obstacles, a flexible formation tracking strategy with obstacle avoidance is necessary. To obtain the optimal performance of formation tracking under actuator constraints and avoid moving obstacles, a formation tracking strategy with adaptive MPC optimization is proposed in this paper. This strategy optimizes the control process of multi-robot formation trajectory tracking and obstacle avoidance, and is implemented in pool experiments. The main contributions of this paper are as follows:

1) To solve the complicated parameter tuning problem in MPC and obtain a faster response to achieve flexible obstacle avoidance, an adaptive model predictive control (AMPC) strategy is established. The adaptive weight matrix is designed in the course of trajectory tracking. The trajectory tracking error is introduced to change the weight matrix adaptively.
2) The collision avoidance of moving obstacles and formation members is realized through the VO method and its application to the constraint design of MPC. The VO uses both relative position and relative speed to avoid obstacles more efficiently.
3) A formation control strategy combining global path planning and local obstacle avoidance is established. The global path planning based on virtual structure method can assign a reasonable global path with time information to each robot. Local tracking and obstacle avoidance control enhances the flexibility during formation movement.

The rest of this paper is organized as follows: In Section 2, the components of multi-USR formation and the experimental platform are introduced. Then, Section 3 describes the problems and introduces the proposed MOAC-AMPC strategy for multi-USRs. Simulation and experimental results are presented in Section 4. Then, the experimental result is discussed in Section 5. Finally, the conclusion is given in Section 6.

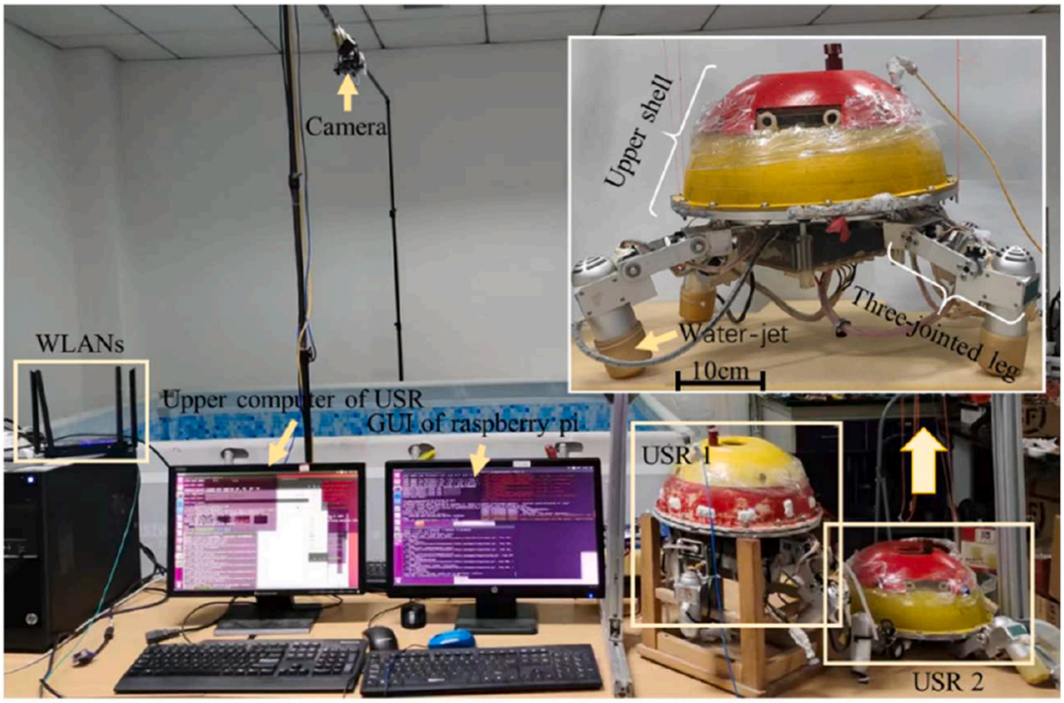

Fig. 2. The main structure of USRs and the multiple USRs platform.

2. The overview of Multi-USR control system

In this section, the kinematics and dynamics models of USR are first presented. Further, the robot structure, sensor arrangement, multi- USR formation and its experimental platform are introduced.

2.1. The USR coordinate system

To accurately describe the mathematical model of multi-USR collaborative formation, the earth-robot coordinate system is shown in Fig. 1.

The pose of the USR in the inertial coordinate system can be expressed as:

$$\boldsymbol{\eta} = [\boldsymbol{\eta}_T, \boldsymbol{\eta}_R] = [x, y, z, \phi, \theta, \psi] \tag{1}$$

where, $\boldsymbol{\eta}_T = [x, y, z]$ is the position of the USR; $\boldsymbol{\eta}_R = [\phi, \theta, \psi]$ is the Euler angle of the USR.

The velocity of the USR in the body coordinate system can be expressed as:

$$\boldsymbol{v} = [\boldsymbol{v}_T, \boldsymbol{v}_R] = [u, v, w, p, q, r] \tag{2}$$

where $\boldsymbol{v}_T = [u, v, w]$ is the linear velocity of the USR translating along the X, Y, and Z axes of the body-frame, and $\boldsymbol{v}_R = [p, q, r]$ is the angular velocity of the USR rotating around the X, Y, and Z axes of the body-frame.

Here, the pose and velocity of the USR are expressed as:

$$\dot{\boldsymbol{\eta}} = J(\boldsymbol{\eta})\boldsymbol{v} \tag{3}$$

where $J(\boldsymbol{\eta}) \in R_{6\times 6}$ is the transition matrix from the body coordinate system to the earth coordinate system.

In USR formation control, to facilitate control, the vector drive system adopted by the USR is X-shaped (the robot's adjacent legs are set at an angle of 90-degree, see Fig. 1). In this X-shaped mode, the structure of the robot in the horizontal plane is symmetrical and its center of buoyancy and center of gravity nearly coincide, so the robot can better maintain its attitude, without rotation in the direction of pitch and roll, and its dynamic model can be simplified to:

$$M\dot{\boldsymbol{v}} + D(\boldsymbol{v})\boldsymbol{v} = \boldsymbol{\tau} \tag{4}$$

where M stands for mass matrix of robot, m = 6.5 kg is the mass of the robot.

$$M = \begin{bmatrix} m & 0 & 0 & 0 \\ 0 & m & 0 & 0 \\ 0 & 0 & m & 0 \\ 0 & 0 & 0 & I_z \end{bmatrix} \tag{5}$$

where I_z represents the inertia tensor matrix about the Z axis associated with the mechanical structure of the robot (Hou et al., 2024).

$D(v)$ represents the matrix associated with hydrodynamic drag.

$$D(v) = D_l + D_{nl}(v) \tag{6}$$

where $D_l, D_{nl}(v)$ represent linear hydrodynamic damping force and second-order nonlinear hydrodynamic damping force, respectively. $\boldsymbol{\tau}$ is for driving force.

2.2. The prototype of USRs

The design of the bionic USR is shown in Fig. 2. The total weight of the robot is 6.7 kg. The upper body mimics the shape of the tortoise's hemispherical shell, which has an overall diameter of 30 cm (Li et al., 2024; Xing et al., 2022; Li and Guo, 2024). The lower half contains a three-jointed leg-like structure that mimics the crawling of a turtle and a water-jet that mimics the propelling of a jellyfish. The multi-joint leg structure design of the robot in our team, on the one hand, meets the design of the robot amphibious movement, and on the other hand, it can freely adjust the angle of the water-jet stream to achieve the underwater vector drive. Please refer to Ref (Xing et al., 2021) for a more detailed description. The driven leg structure improves the flexibility and maneuverability of USRs.

Circuits, control modules, and various sensors, such as Inertial Measurement Units (IMUs), pressure sensors, UAV cameras, and speed sensors, are integrated into the robot's upper hemispherical shell to

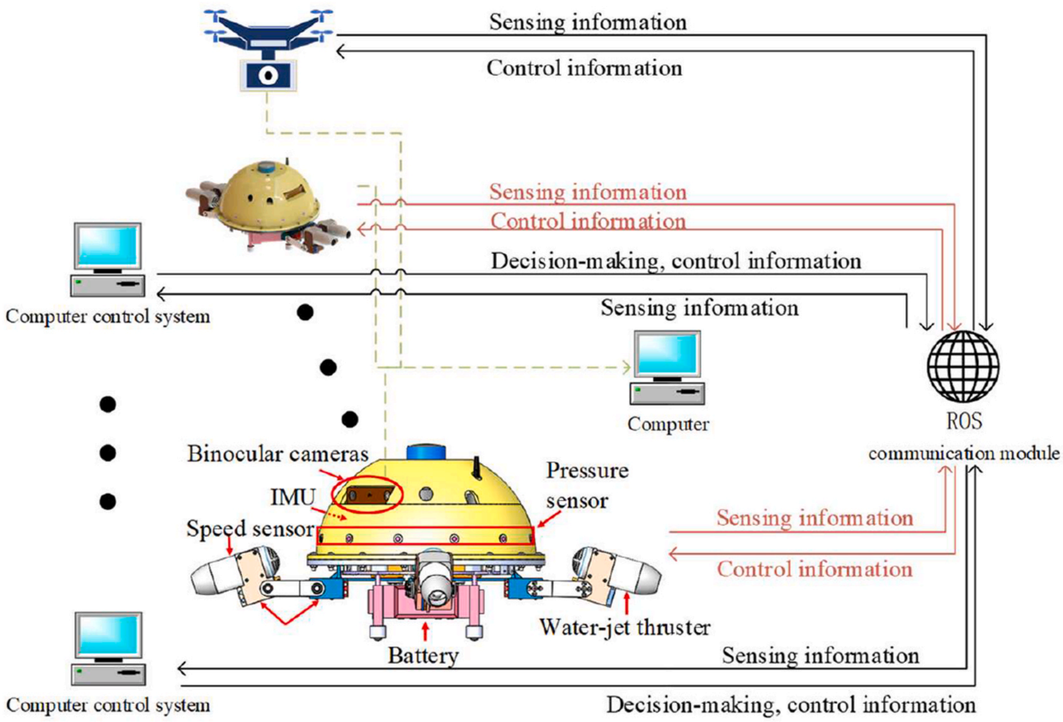

Fig. 3. The communication structure of multiple USRs platform.

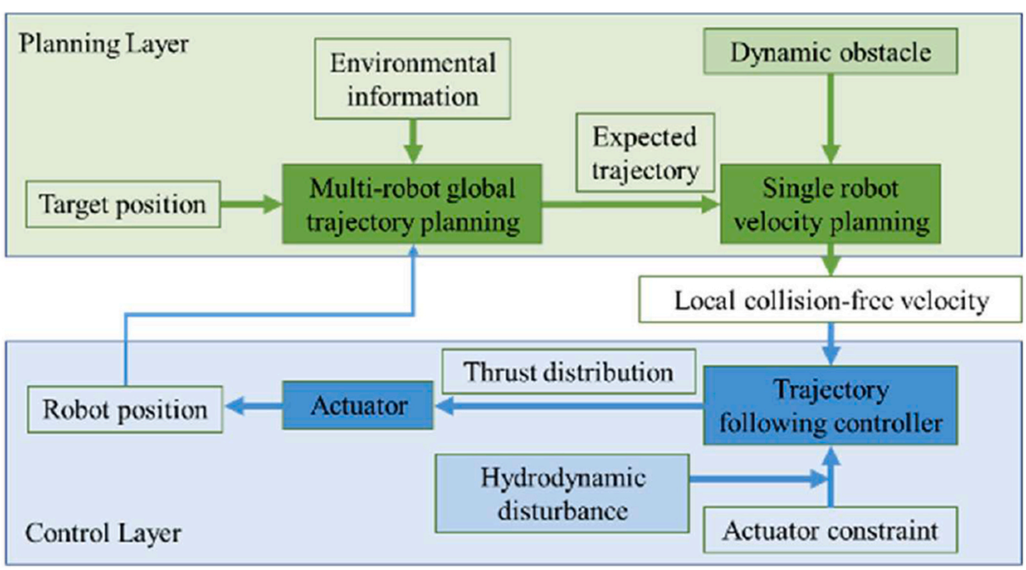

Fig. 4. The planning and control of USR formation with X-shaped (see Fig. 1) vector drive system.

sense information about the external environment. The UAV camera information processing relies on NVIDIA Jetson Nano module. For the binary images obtained after processing, morphological image processing is used to obtain a relatively ideal contour map, and template matching is performed. IMU adopts the LORD Sensing 3DM-GX5 series industrial inertial navigation system to predict the posture, motion speed and position of the robot through the posture and acceleration changes. The speed sensors mainly measure the propeller speed through the current feedback, which can be used to correct the measurement of robot speed by IMU. Pressure sensors are used to sense underwater depth. IMU and speed sensors are used to verify and optimize the camera's measurement results.

The multi-USR platform shown in Fig. 2 consists of multiple USRs, an Unmanned Aerial Vehicle (UAV), and a computer control system. The camera attached to the UAV provides location information to the USRs and obstacles. The Kalman filter is used to predict the position and velocity of the robot. The principle is shown in Appendix A. In shallow sea environments, the combination of UAV and underwater robots can effectively provide the relative position relationship between robots and the position of obstacles, which is conducive to formation and collaborative obstacle avoidance.

The computer control system is the brain of robots and is responsible for processing perceptual information and task requirements to produce appropriate instructions. The underwater acoustic communication module is used for communication in ocean environments. In pool experiments, to reduce the communication error and focus on the research

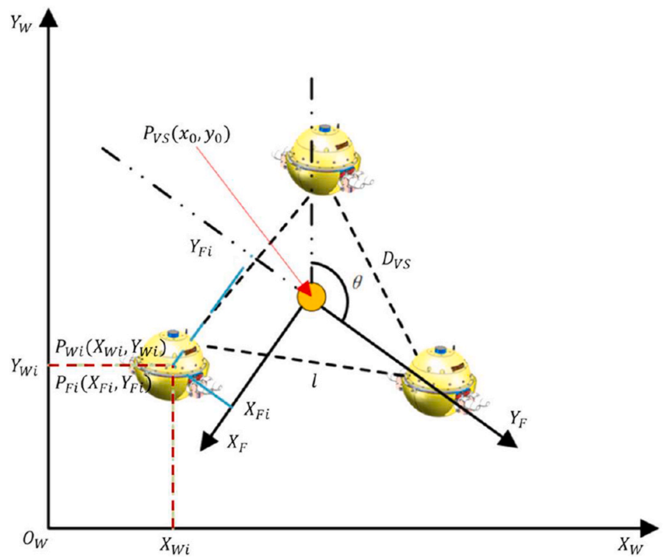

Fig. 5. The schematic diagram of formation control strategy under the virtual structure method.

of formation obstacle avoidance in this paper, the communication structure is mainly realized through optical fiber module, local area network and ROS communication module, as shown in Fig. 3. The USRs and the UAV communicate via the ROS multi-robot communication module. In the case of short-distance optical fiber transmission, there is almost no communication failure. The optical fiber connections ensure reliable communication. The planning and control based on task decomposition is shown in Fig. 4.

2.3. The multiple USRs formation with virtual structure method

The virtual structure method uses a reference point as a leader, from which the desired track points for all formation members are determined. In this paper, this method is mainly used to control the global position information of the robots, so the USRs still have greater freedom in other aspects.

The virtual leader's track points are provided by a predefined reference track. The trajectory points of formation members are obtained from formation shape parameters. In Fig. 5, O_W is the origin of the world coordinate system. $P_{Wi} = (X_{Wi}, Y_{Wi})$ represents the position of robot *i* in the earth coordinate system, assuming that the robot is labeled *1, 2, …, n*. P_{VS} (x_0, y_0) represents the position of the formation center in the earth coordinate system. D_{VS} represents the side length of the virtual triangle structure. θ is the forward heading angle of the formation under the virtual structure. Y_F represents the moving direction of the robot and is also the positive direction of the *Y-axis* of the formation reference frame. X_F is the positive *X-axis* of the formation reference frame. The robot vectors in the formation reference coordinate system are $P_{Fi} = (X_{Fi}, Y_{Fi})$ $i = 1 \ldots n$. The coordinate transformations from the vector P_F to P_W require rotation and translation. The rotation matrix R and the translation matrix T are as follows. The robot has a strong resilience to keep from turning, so the rotation is in the horizontal plane.

$$R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{7}$$

$$T = (x_0, y_0, z_0) \tag{8}$$

The steps of the virtual structure method are as follows: Firstly, the position of each robot member in the formation is obtained, and the coordinates of the virtual structure point in the formation are calculated. Secondly, the coordinates of virtual structure points expected at the next time are updated according to the expected speed and step size. Finally, the control algorithm is invoked to control the speed and position of the robot to make the robot track the desired trajectory.

3. The moving obstacles avoiding and formation control strategy

The virtual structure method is used to plan the trajectory of the USRs formation globally. Then, the trajectory tracking and obstacle avoidance of the robots are realized by the improved MPC trajectory tracking algorithm and the moving obstacle avoidance and constraint strategy locally. In resource exploration and maintenance work in shallow sea, the overall multi-USR formation control system is shown in

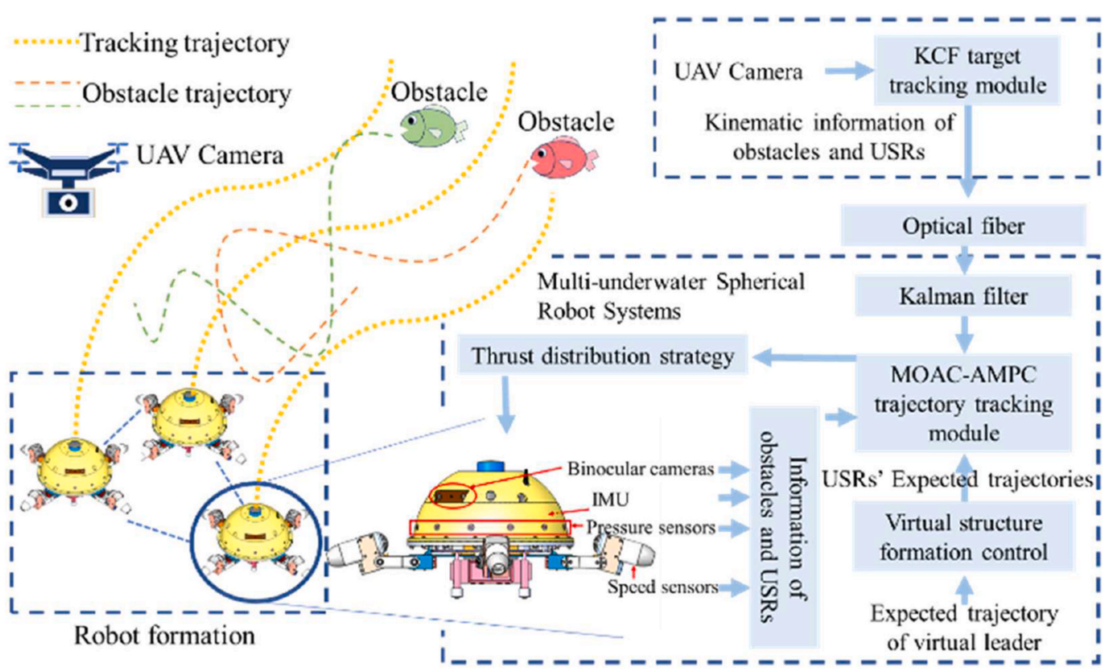

Fig. 6. The diagram of the Multiple USRs control system.

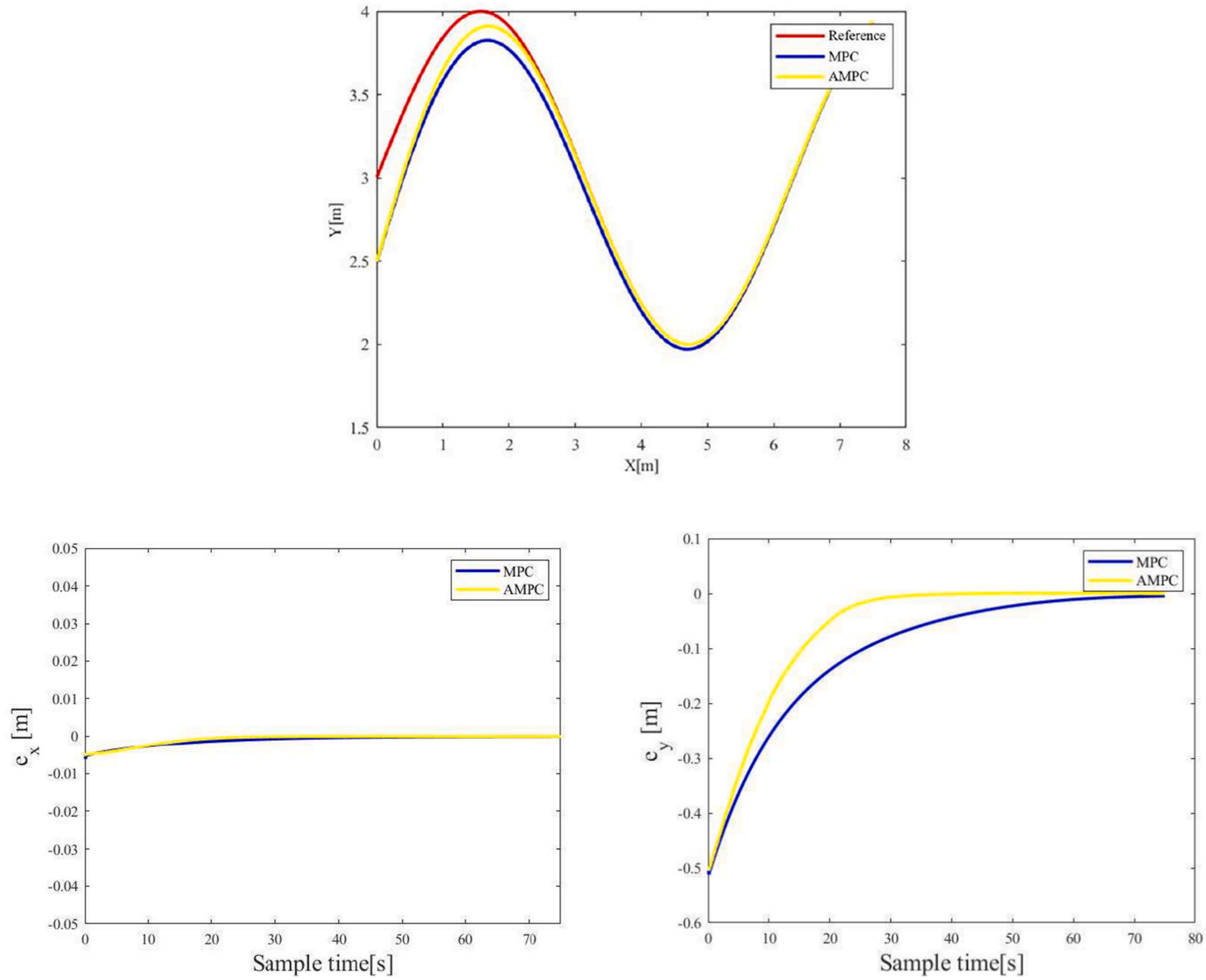

Fig. 7. The comparison of the AMPC and MPC algorithms. (a) The motion trajectory (b) Tracking errors in the X direction (c) Tracking errors in the Y direction.

Table 1
The Mean Square Errors (MSEs) for the AMPC and the MPC trajectory.

Direction	MSEs of AMPC (cm)	MSEs of MPC (cm)	Reduction of MSEs
X	$2.66*10^{-3}$	$3.03*10^{-3}$	12.2 %
Y	1.85	2.65	30.2 %

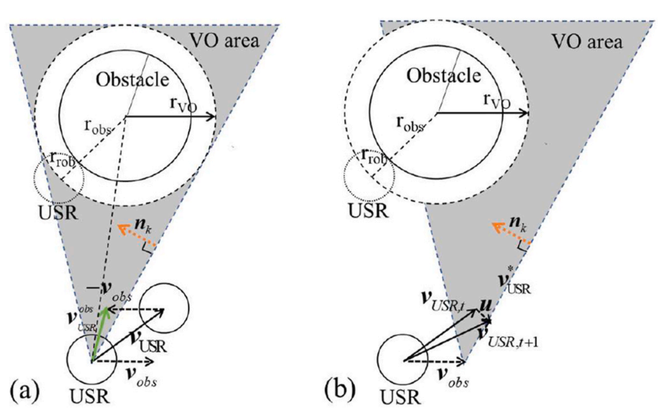

Fig. 8. The schematic diagram of the VO method. (a) The velocity increment calculated from the velocity of the USR relative to the obstacle. (b) The change in the absolute velocity of the USR.

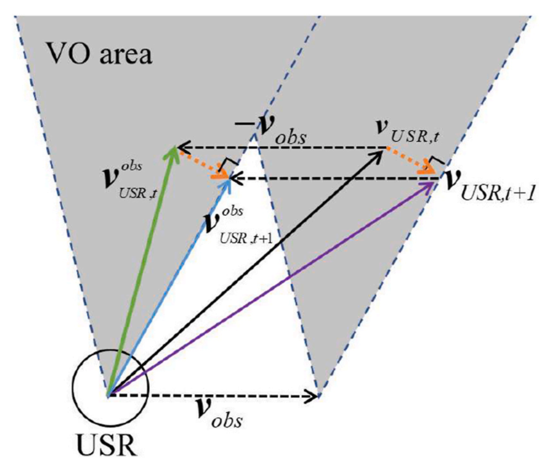

Fig. 9. The velocity constraint.

Fig. 6. The dotted yellow lines represent the planned trajectories. The advantages of this design of multi-USR formation control system are: First, the inflexible structure of virtual structure method in the obstacle avoidance process is reduced. Otherwise, when one robot avoids obstacles, other robots will deviate from their own trajectory. In this way, other formation members are not affected by obstacle avoidance

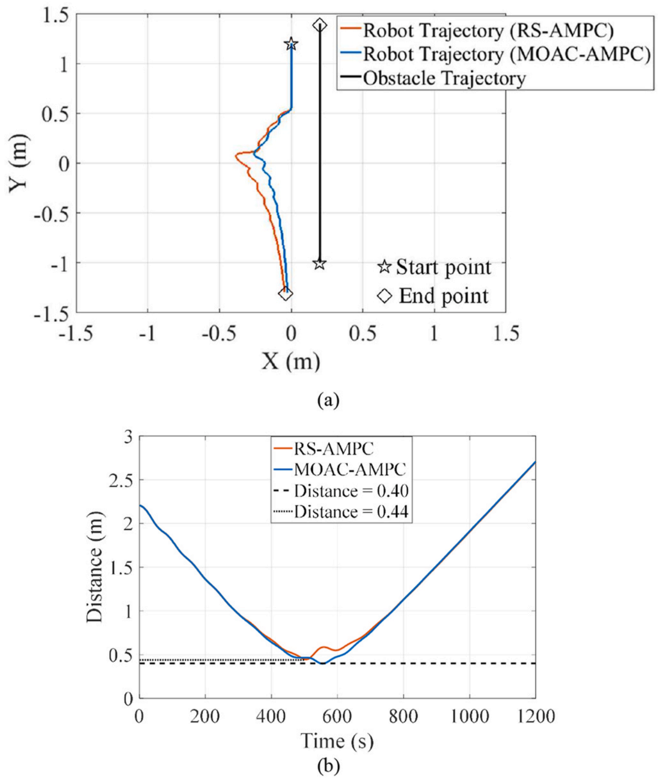

Fig. 10. The comparison of RS-AMPC and MOAC-AMPC algorithms. (a) The obstacle avoidance trajectory. (b) The distances between robots and obstacles.

members, but along the globally planned trajectory. This may reduce the robots' excess affected obstacle avoidance movements and fluctuations, which can reduce energy consumption.

Then, most underwater obstacles are not stationary and their motion state is difficult to predict, local obstacle avoidance based on velocity obstacle method enhances the flexibility of obstacle avoidance. In addition, the trajectory of global path planning made by the virtual structure method is time-dependent and has the alignment of formation. The formation can be formed when every robot tracked the trajectory.

In the following, the MPC trajectory tracking algorithm is improved, aiming to speed up the tracking process and reduce the time required to track to the trajectory. The obstacle avoidance constraint matrix is designed based on the VO method to avoid obstacles while tracking the trajectory.

3.1. The improved formation trajectory tracking method

To enhance the robustness and efficiency of the MPC method, an adaptive weight matrix adjustment method based on output error is designed. Suppose that the system state model of the i-th USR in the formation tracking problem obtained from Equations (4)–(6) is as follows:

$$\dot{x}_i = \begin{bmatrix} J(\eta)v_i \\ M_i^{-1}\left(\tau_i - D_l v_i - D_{nl}|v_i|^2\right) \end{bmatrix} = f(x_i, \tau_i) \tag{9}$$

For the i-th USR, $\boldsymbol{x}_i = \mathrm{col}(\boldsymbol{\eta}_i, \boldsymbol{v}_i) \in \mathbb{R}^6$ is the state vector, $\boldsymbol{\eta}_i$ is the position vector, $\boldsymbol{v}_i$ is the velocity vector, and $\boldsymbol{\tau}_i \in \mathbb{R}^3$ is the control input. The reference trajectory $\boldsymbol{x}_{ref,k+1}$ is used to create the new state $\boldsymbol{\xi}_{k+1}$, as shown in Equation (10).

$$\begin{aligned} \boldsymbol{\xi}_{k+1} &= A\boldsymbol{\xi}_k + B\Delta\tau_k \\ \eta_k &= C\boldsymbol{\xi}_k \\ \boldsymbol{\xi}_{k+1} &= \boldsymbol{x}_{k+1} - \boldsymbol{x}_{ref,k+1} \end{aligned} \tag{10}$$

The discrete matrices A and B are described below:

$$\boldsymbol{A} = \begin{bmatrix} I + TA_t & TB_t \\ 0_{m\times n} & I_m \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} TB_t \\ I_m \end{bmatrix}, \boldsymbol{C} = \boldsymbol{I}_{2\times 2} \tag{11}$$

where the linearization parameter is

$$A_t = \begin{bmatrix} \boldsymbol{0}_{2\times 2} & I_2 \\ \boldsymbol{0}_{2\times 2} & \boldsymbol{M}^{-1}(-\boldsymbol{D}_l - 2\boldsymbol{D}_{nl}|\boldsymbol{v}|) \end{bmatrix}, B_t = \begin{bmatrix} \boldsymbol{0}_{2\times 2} \\ \boldsymbol{M}^{-1} \end{bmatrix} \tag{12}$$

To establish a standard quadratic problem, Hessian matrix $\boldsymbol{H}_k$ and gradient matrix $\boldsymbol{G}_k$ are designed. Λ and ε are added to the secondary optimization to prevent the occurrence of no solution.

$$\begin{aligned} &\min J_k = \left[\Delta\boldsymbol{\tau}_k^T, \varepsilon\right]\boldsymbol{H}_k\left[\Delta\boldsymbol{\tau}_k^T, \varepsilon\right]^T + \boldsymbol{G}_k\left[\Delta\boldsymbol{\tau}_k^T, \varepsilon\right]^T \\ &\text{s.t. } \Delta\boldsymbol{\tau}_{\min} \le \Delta\boldsymbol{\tau}_k \le \Delta\boldsymbol{\tau}_{\max}, \boldsymbol{A}_{uneq}\Delta\boldsymbol{\tau}_k \le \boldsymbol{b}_{uneq} \end{aligned} \tag{13}$$

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{\Theta}^T\boldsymbol{Q}\boldsymbol{\Theta} + \boldsymbol{R} & \boldsymbol{0} \\ \boldsymbol{0} & \Lambda \end{bmatrix} \quad \boldsymbol{G} = 2(\boldsymbol{\Phi}\boldsymbol{\xi})^T\boldsymbol{Q}\boldsymbol{\Theta} \tag{14}$$

$$\Phi = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^N \end{bmatrix}, \Theta = \begin{bmatrix} CB \\ CAB \\ \vdots \\ C\left(\sum_{i=1}^{N-1} A^i\right)B \end{bmatrix} \tag{15}$$

The coefficient matrix $\boldsymbol{A}_{uneq}$ of the control increment in the control constraint is as follows:

$$\boldsymbol{A}_{uneq} = \begin{bmatrix} \boldsymbol{\Lambda}_{down,Nc} \otimes \boldsymbol{I}_{Nu} & \boldsymbol{0}_{Nu\cdot Nc,1} \\ -\boldsymbol{\Lambda}_{down,Nc} \otimes \boldsymbol{I}_{Nu} & \boldsymbol{0}_{Nu\cdot Nc,1} \end{bmatrix} \tag{16}$$

where $\boldsymbol{\Lambda}_{down,Nc}$ $\boldsymbol{b}_{uneq}$ is as follows:

$$\boldsymbol{\Lambda}_{down,Nc} = \begin{bmatrix} 1 & & \\ \vdots & \ddots & \\ 1 & \dots & 1 \end{bmatrix}_{Nc} \tag{17}$$

$$\boldsymbol{b}_{uneq} = \begin{bmatrix} \boldsymbol{I}_{Nc,1} \otimes \boldsymbol{\tau}_{\max} - \boldsymbol{I}_{Nc,1} \otimes \boldsymbol{\tau}_k \\ -\boldsymbol{I}_{Nc,1} \otimes \boldsymbol{\tau}_{\min} + \boldsymbol{I}_{Nc,1} \otimes \boldsymbol{\tau}_k \end{bmatrix} \tag{18}$$

In a multivariable system, the effects of parameters on different variables are coupled, which increases the parameter tuning complexity of the MPC. To solve this problem, an adaptive weight matrix adjustment method based on tracking error is designed. The reference trajectory is:

$$\boldsymbol{Y}_{\boldsymbol{ref},k} = \left[\boldsymbol{\eta}_{ref,k}, \dots, \boldsymbol{\eta}_{ref,k+Np}\right]^T \tag{19}$$

The output error of the system satisfies:

$$\boldsymbol{Err}_k = \frac{\sum_{j=1}^{Np}\left[\boldsymbol{Y}_{k+j} - \boldsymbol{Y}_{\mathrm{ref},k+j}\right]}{Np} \tag{20}$$

The weight matrix is designed as follows:

$$\boldsymbol{Q}_{ada,k} = K_{ada}\boldsymbol{Err}_k \tag{21}$$

where K_{ada} is a constant parameter. The parameter K_{ada} of the weight matrix needs to obtain a suitable value, a small parameter value will lead to poor tracking effect, and a large value will lead to unstable motion. In this paper, $K_{ada} = 10$. Standardized $\boldsymbol{Q}_{normal,k}$ is used to balance the control effects of various state variables.

$$\boldsymbol{Q}_{normal,k} = \frac{1}{1 + e^{-\boldsymbol{Q}_{ada,k}+2.5}} \tag{22}$$

The comparison of the AMPC and MPC algorithms is shown in Fig. 7. The tracking errors in the X direction are very small, and there is no obvious difference between the two algorithms. The tracking errors of AMPC in the Y direction are smaller than that of MPC. Compared with MPC, AMPC can achieve similar effect for trajectory tracking with small

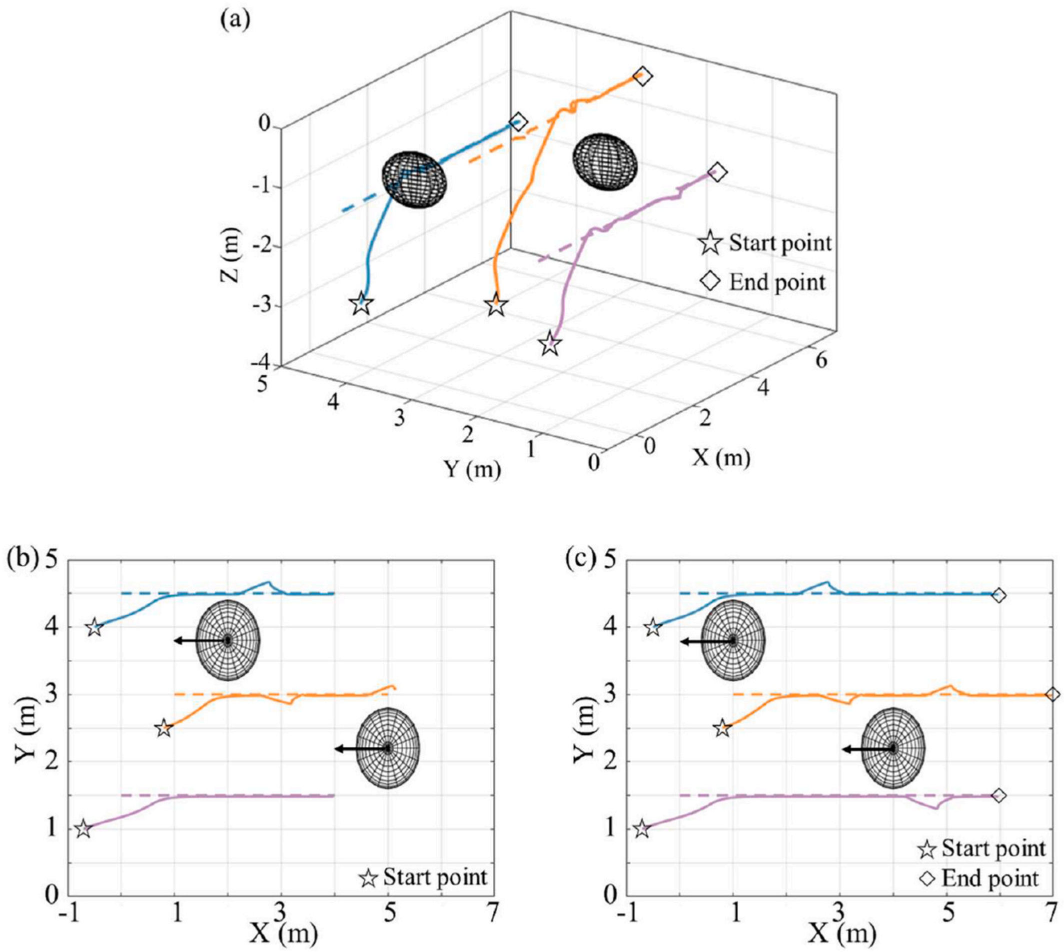

Fig. 11. The simulation verification of the robot avoiding moving obstacles. (a) Three-dimensional diagram. (b) XY plane diagram at t1. (c) XY plane diagram at t2.

Table 2
The robots and obstacle information.

Object	Radius (m)	Start point (m)	End Point (m)	Velocity (m/s)
Robot-1	0.30	(-2.00, 5.00)	(6.00, 2.70)	time varying
Robot-2	0.30	(-2.00, 3.00)	(5.00, 4.70)	time varying
Robot-3	0.30	(-1.50, 2.00)	(5.00, 0.70)	time varying
Obstacle-1	0.30	(4.00, 2.00)	(-0.80, 2.00)	(-0.80, 0.00)
Obstacle-2	0.30	(6.00, 5.00)	(1.20, 5.00)	(-0.80, 0.00)

initial error. For the track tracking with large initial error, AMPC can track the target track faster and reduce the tracking time through parameter adaptive adjustment. In the weight matrix, the weights in the X direction and the Y direction are equal. The AMPC adaptively adjusts the weight matrix according to the errors in the Y direction to reduce the errors.

The percentage reduction of the Mean Square Errors (MSEs) for the AMPC compared with MPC trajectory in X and Y directions are 12.2 % and 30.2 % respectively, shown in Table 1. Noted that, the control input and control increment are within the constraint range.

3.2. The Moving Obstacle Avoidance Constraint design

The Moving Obstacle Avoidance Constraint (MOAC) was designed based on the Velocity Obstacle (VO) method. In Fig. 8, r_{obs} represents the radius of the obstacle, and r_{rob} represents the radius of the robot. r_{VO} represents the expanded radius of the obstacle, thus forming the area VO where a collision may occur. v_t represents the current velocity of the robot. v_{t+1} represents the speed of the robot to avoid obstacles at $t+1$ moment. v_{obs} represents the velocity of obstacles, and u represents the velocity increment. n represents the normal vector of the nearest boundary to escape VO.

Then, the nearest boundary is selected to calculate the normal vector n and the velocity increment u. Finally, the velocity of the USR is calculated. The green arrow in Fig. 9 shows the robot's velocity relative to the obstacle, falling on the right half of the VO area. The starting point of the velocity increment escaping the VO boundary (orange arrow in Fig. 9) is the endpoint of the robot's velocity relative to the obstacle. The endpoint is the projection of the robot's velocity relative to the obstacle to the VO boundary. The solution v_{t+1} for the absolute velocity of the robot is shown in the purple line in Fig. 9.

Assume that the velocity of the obstacle at the predicted horizon is a constant. The velocity constraint inequality can be described as:

$$(\boldsymbol{v}_{k+1} - (\boldsymbol{v}_k + \delta\boldsymbol{v}_k))\boldsymbol{n} < 0, k = 1 \dots Np \tag{23}$$

where $\delta\boldsymbol{v}_k$ represents the increment in velocity. Here, the dynamic constraint transforms to $\boldsymbol{N}_{VO,k}(\boldsymbol{V}_{k+1} - \boldsymbol{V}_k - \delta\boldsymbol{V}_k) < 0$.

To obtain the velocity vector in the state vector, a normal vector matrix containing zero vector is designed.

$$\boldsymbol{N}_{VO,k} = \begin{bmatrix} \boldsymbol{n}_{z1} & \boldsymbol{0}_{1\times4} & \dots & \boldsymbol{0}_{1\times4} \\ \boldsymbol{0}_{1\times4} & \boldsymbol{n}_{z2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \boldsymbol{0}_{1\times4} \\ \boldsymbol{0}_{1\times4} & \dots & \boldsymbol{0}_{1\times4} & \boldsymbol{n}_{zNp} \end{bmatrix} \tag{24}$$

where $\boldsymbol{n}_{zi} = [\boldsymbol{0}_{1\times2} \quad \boldsymbol{n}_i]$, and $\boldsymbol{N}_{VO,k}\boldsymbol{V}_{k+1}$ can be obtained from the properties of normal vector matrices.

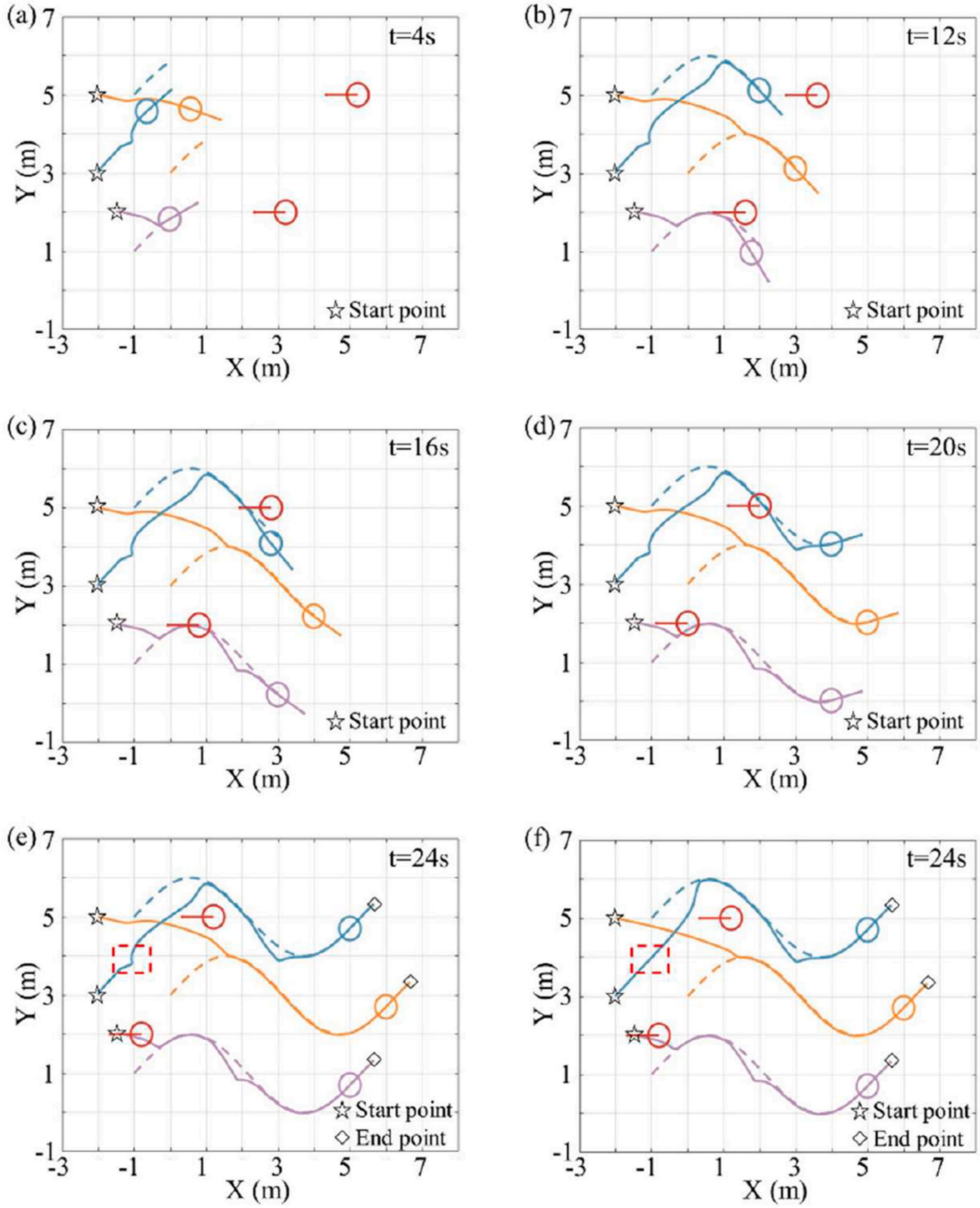

Fig. 12. The comprehensive experiment of moving obstacle avoidance in formation. (a) t = 4s. (b) t = 12s. (c) t = 16s. (d) t = 20s. (e) t = 24s with the collision avoidance of the formation members. (e) t = 24s without the collision avoidance of the formation members.

$$\boldsymbol{N}_{VO,k}\boldsymbol{V}_{k+1} = \boldsymbol{N}_{VO,k}\boldsymbol{\eta}_{k+1} \tag{25}$$

where $\boldsymbol{\eta}_{k+1}$ are calculated by controlling increment $\boldsymbol{\tau}_k$, which provides a transformation from velocity constraint to control increment constraint.

$$\boldsymbol{\eta}_{k+1} = \Phi\boldsymbol{\eta}_k + \Theta\Delta\boldsymbol{\tau}_k \tag{26}$$

Since the control increment is the target quantity to be solved, the output state $\boldsymbol{\eta}_{k+1}$ at $k+1$ is unknown. It is important to note that the final predicted output state $\boldsymbol{\eta}_k$ is not computed by $\Delta\boldsymbol{\tau}_{k-1}$, but by the unknown vector $\Delta\boldsymbol{\tau}_k$. Therefore, the output state $\boldsymbol{\eta}_0$ at the current time is needed.

$$\boldsymbol{\eta}_{0,k} = \begin{bmatrix} \boldsymbol{x}_k^T & \mathbf{0}_{1\times 4(Np-1)} \end{bmatrix}^{\mathrm{T}} \tag{27}$$

The velocity constraint inequality is expanded in the prediction time domain, and the inequality including control increment is obtained.

$$\begin{aligned}&\boldsymbol{N}_{VO,k}(\boldsymbol{\Phi}_k\boldsymbol{\eta}_k + \boldsymbol{\Theta}_k\Delta U_k) < \\ &\boldsymbol{N}_{VO,k}\left(\left(\boldsymbol{I}_{last}(\boldsymbol{\Phi}_k\boldsymbol{\eta}_k + \boldsymbol{\Theta}_k\Delta U_k) + \boldsymbol{\eta}_{0,k}\right) + \delta\boldsymbol{V}_k\right)\end{aligned} \tag{28}$$

To make sure it corresponds to the dimensions of $\boldsymbol{N}_{VO,k}$, $\delta\boldsymbol{V}_{k+1}$ is designed to be

$$\delta\boldsymbol{V}_k = \begin{bmatrix} \mathbf{0}_{1\times 2} & \delta\boldsymbol{v}_1{}^{\mathrm{T}} \ldots \mathbf{0}_{1\times 2} & \delta\boldsymbol{v}_{Np}{}^{\mathrm{T}} \end{bmatrix} \tag{29}$$

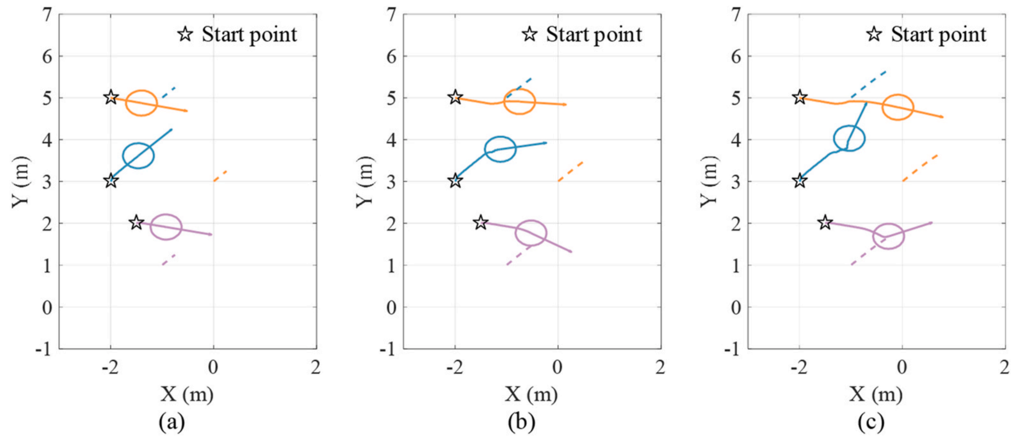

Fig. 13. The comprehensive experiment of moving obstacle avoidance in formation. (a) t = 1s. (b) t = 2s. (c) t = 3s.

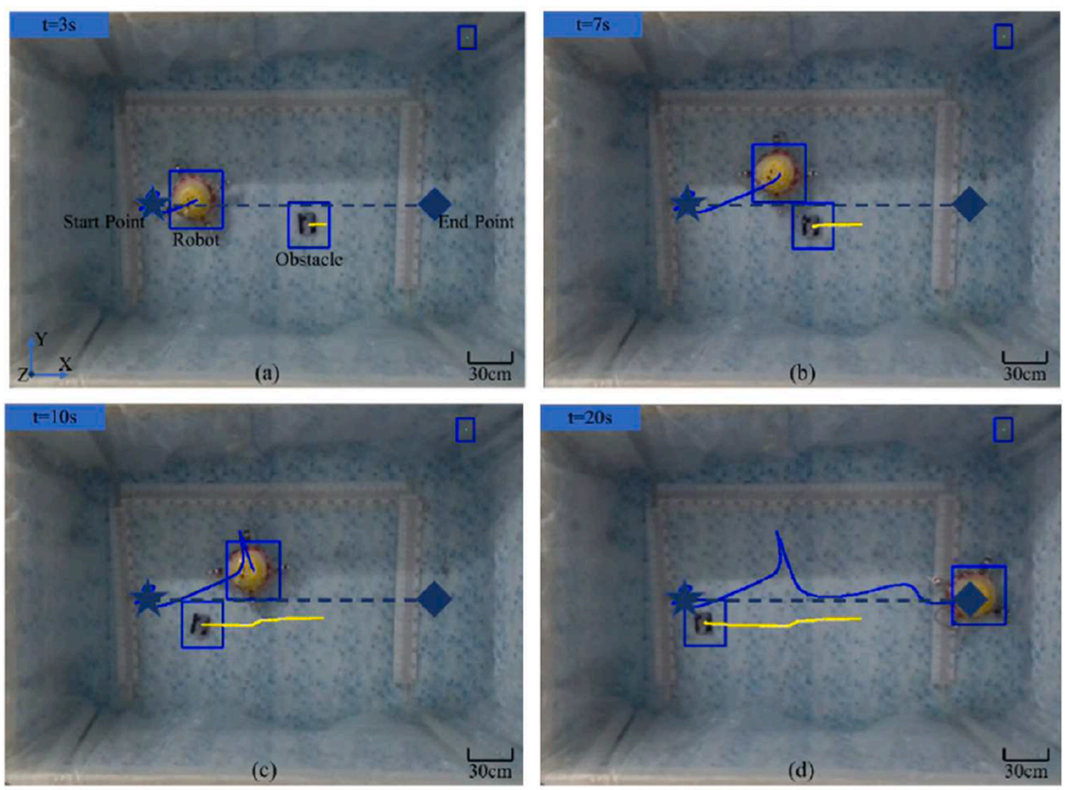

Fig. 14. The moving obstacle avoidance experiment of a single robot.

Table 3
Moving obstacles and robot information.

The start point of obstacle (cm)	The end point of obstacle (cm)	The start point of robot (cm)	The end point of robot (cm)
(145.0, 104.0)	(57.0, 110.0)	(40.0, 100.0)	(250.0, 100.0)

The matrix coefficients of control increment are extracted and the obstacle avoidance control increment constraint is obtained.

$$\boldsymbol{A}_{VO} = \boldsymbol{N}_{VO,k}(\boldsymbol{\Theta}_k - \boldsymbol{I}_{last}\boldsymbol{\Theta}_k) \tag{30}$$

Considering the effect of the relaxation factor, it is necessary to add the zero vector to the coefficient matrix.

$$\boldsymbol{A}_{VO} = \begin{bmatrix} \boldsymbol{N}_{VO,k}(\boldsymbol{\Theta}_k - \boldsymbol{I}_{last}\boldsymbol{\Theta}_k) & \boldsymbol{0}_{Np\times 1} \end{bmatrix} \tag{31}$$

$\boldsymbol{b}_{VO}$ is calculated as:

$$\boldsymbol{b}_{VO} = \boldsymbol{N}_{VO,k}\left(\boldsymbol{I}_{last,k}\boldsymbol{\Phi}_k\boldsymbol{\eta}_k - \boldsymbol{\Phi}_k\boldsymbol{\eta}_k + \boldsymbol{\eta}_{k,0} + \delta\boldsymbol{V}_k\right) \tag{32}$$

The final form of an optimization problem that considers all obstacles can be solved with a linear constrained objective function solver, as shown in Equation (33).

$$\begin{aligned} &\min J_k = \left[\Delta\boldsymbol{\tau}_k^T, \varepsilon\right]\boldsymbol{H}_k\left[\Delta\boldsymbol{\tau}_k^T, \varepsilon\right]^T + \boldsymbol{G}_k\left[\Delta\boldsymbol{\tau}_k^T, \varepsilon\right]^T \\ &\text{s.t.} \quad \Delta\boldsymbol{\tau}_{\min} \le \Delta\boldsymbol{\tau}_k \le \Delta\boldsymbol{\tau}_{\max} \\ &\begin{bmatrix} \boldsymbol{A}_{uneq} \\ \boldsymbol{A}_{VO,1} \\ \vdots \\ \boldsymbol{A}_{VO,Nobs} \end{bmatrix} \Delta\boldsymbol{\tau}_k < \begin{bmatrix} \boldsymbol{b}_{uneq} \\ \boldsymbol{b}_{VO,1} \\ \vdots \\ \boldsymbol{b}_{VO,Nobs} \end{bmatrix} \end{aligned} \tag{33}$$

Optimization problems can be solved by quadratic objective function solvers with linear constraints. It can be seen that the obstacle avoidance

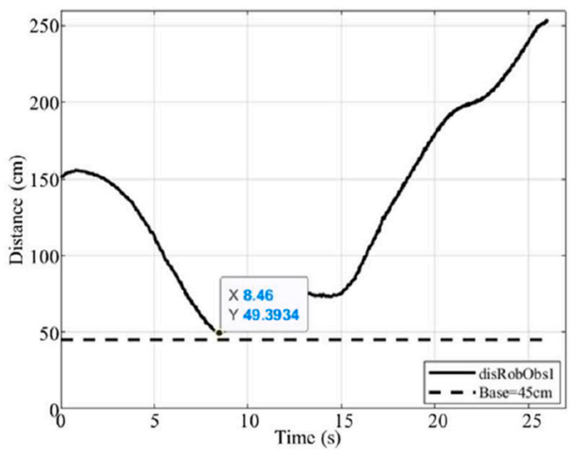

Fig. 15. The distance between the robot and the obstacle in the obstacle avoidance experiment.

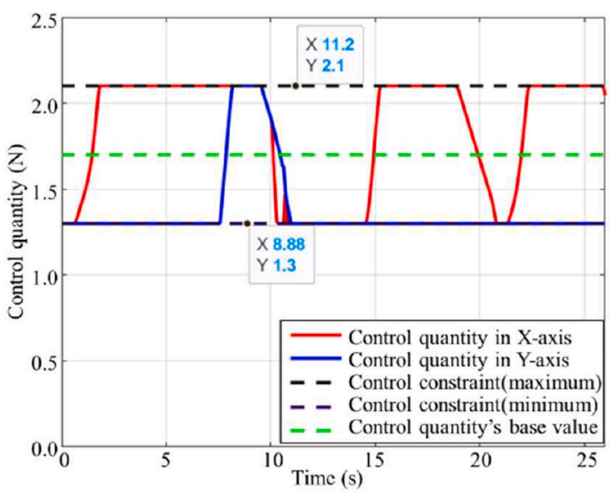

Fig. 16. The control inputs of the robot in the X and Y axis.

strategy is suitable for robot systems that can transform velocity constraint into control quantity constraint through dynamic model. And the obstacle avoidance strategy should be used as a constraint term of MPC or quadratic optimization problems.

Here, two obstacle avoidance strategies are designed based on AMPC for comparison. The first is as shown in Fig. 9 which combined with MOAC-AMPC to obtain obstacle avoidance velocity and obstacle avoidance displacement. The other is to randomly select the obstacle avoidance velocity that does not fall within the VO obstacle avoidance boundary, and this strategy is labeled as RS-AMPC. To illustrate the effectiveness of the obstacle avoidance algorithm more clearly in the figure, a single robot is selected for obstacle avoidance experiments. The radius of the robot is 30.0 cm, the starting point is (0.0, 1.2) m, the end point is (0.0, −1.2) m, and the velocity is −0.1 m/s. The radius of the obstacle is 10.0 cm, the starting point is (0.2, −1.0) m, the end point is (0.2, 1.4) m, and the velocity is 0.1 m/s.

It can be seen from Fig. 10 that the minimum obstacle avoidance distances of RS-AMPC and MOAC-AMPC are 0.44 m and 0.40 m, respectively. Therefore, in terms of performance, the unnecessary moving distance of the robot is reduced in the obstacle avoidance process, and the reduction is 9 %. MOAC-AMPC (solid blue line) has a better obstacle avoidance effect, and the robot is closer to the expected trajectory while staying away from the obstacle (solid black line). Both of them are greater than the sum of USR and the obstacle radius (0.40 m). In addition, the calculation time consumption of the combination of the improved obstacle avoidance strategy and the AMPC is almost the same as that before the improvement.

4. Simulations and experiments

Two simulation scenarios and two experimental scenarios were set up to verify the proposed MOAC-AMPC strategy. The motion of USRs in the first simulation scenario showed the obstacle avoidance effect of the proposed strategy by avoiding moving obstacles without replanning the trajectory. The second simulation scenario verified the strategy on avoiding both obstacles and robots in formation. The first experimental scenario verified the feasibility of avoiding obstacles. The second experimental scenario verified the multi-USR formation trajectory tracking in the presence of moving obstacles under the proposed strategy.

4.1. Simulation results

Firstly, the proposed obstacle avoidance strategy is verified based on the dynamic model mentioned earlier. The obstacle avoidance effect is shown in Fig. 11. The dashed line represents the desired trajectory of the robot, and the solid line represents the actual trajectory. Different robot trajectories are distinguished by different trajectory colors. There are two obstacles moving in the negative direction of the X axis. The obstacle avoidance threshold of the VO method is set as 1.2 m. As the radius of the robot is 0.3m and the radius of the obstacle is set as 0.9m. In the virtual structure method, the robot position can be transformed into the relative coordinate representation, and the center of the virtual structure is the origin. Orange represents robot 1 and the relative coordinates are (0.0, 0.0, 0.0) m. Blue represents robot 2 with relative coordinates of (−1.0, 1.5, −1.0) m. Purple represents robot 3, whose relative coordinates are (−1.0, −1.5, −1.0) m. The robot follows the trajectory while avoiding obstacles. Fig. 11(b) and (c) show the robot's position state at different moments, which verifies the timeliness requirement of the algorithm.

Next, the collision avoidance ability of the robot formation members was verified. On the basis of verifying the computational capability of the algorithm in the three-dimensional space in Fig. 11, obstacle avoidance is returned to the two-dimensional plane to analyze the obstacle avoidance process more clearly and intuitively. The experiment scenario is that three robots start from any point and track the desired trajectory while avoiding obstacles and the robot formation members. Robot and obstacle information are shown in Table 2. The effect of cooperative obstacle avoidance of robot formation is shown in Fig. 12. Similarly, the dotted line represents the desired trajectory of each robot, and the solid line represents the actual trajectory. The orange circle represents *Robot-1*, and the orange line represents its trajectory. Blue and purple represent *Robot-2* and *Robot-3*, respectively. Obstacles are represented by red circles. The straight segments in the circles represent the velocity direction of the robots or obstacles at this moment.

The reference trajectory of the robot center is described as

$$\begin{cases} p_x = 0.1kT \\ p_y = 3 + \sin(0.1kT) \end{cases}, k = 1\ldots1500 \tag{34}$$

The solid blue line in Fig. 12(a) is curved, representing *Robot-2* (blue) dodging *Robot-1* (orange). *Robot-3* (purple) in Fig. 12(b) is avoiding *Obstacle-1* in the bottom half. Robot 2 in Fig. 12(c) is dodging *Obstacle-2* in the top half. In Fig. 12(d), the three robots return to their respective expected trajectories, and in Fig. 12(e), the robots arrive at the end. The comparison between Fig. 12(f) and (e) aims to illustrate the obstacle avoidance function between the members of the multi-robot formation. In Fig. 12(e), the trajectory of the blue robot with the collision avoidance strategy of formation members is bent at the initial stage. That of the blue robot without collision avoidance strategy of formation members is not bent in Fig. 12(f). The effectiveness of the collision avoidance

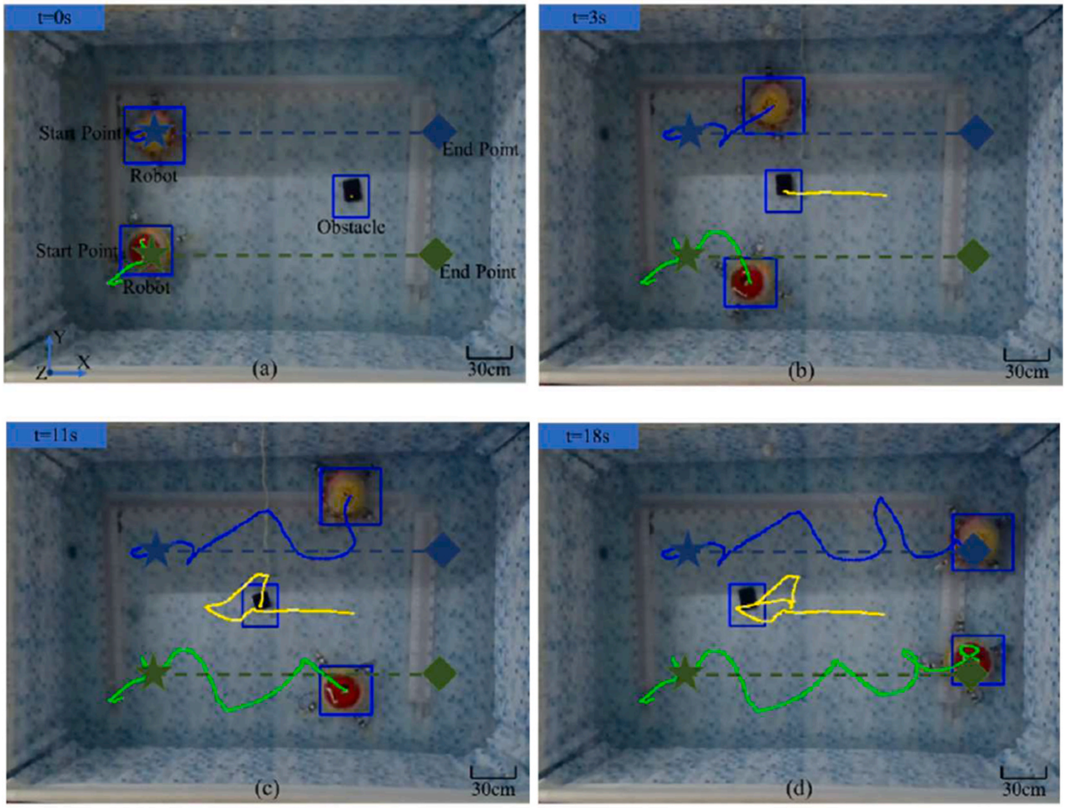

Fig. 17. The comprehensive experiment of collaborative formation and obstacle avoidance.

Table 4
The obstacles and robot information.

Point	Robot-1 (yellow) (cm)	Robot-2 (Red) (cm)	Obstacle (cm)
Start point	(19.0, 169.0)	(29.0, 65.0)	(215.0, 114.0)
End point	(297.0, 165.0)	(281.0, 83.0)	(95.0, 122.0)

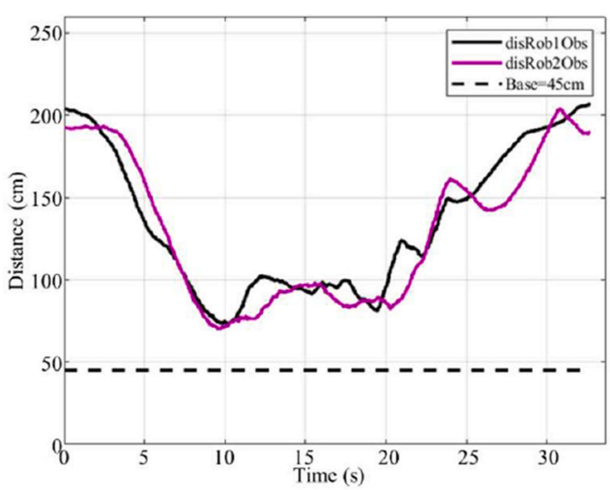

Fig. 18. The distance between the robot and the obstacle in the formation obstacle avoidance experiment.

strategy of formation members is proved in simulations.

To further explain the collision avoidance process between robots and the collision avoidance principle, the collision avoidance process of 1–3 s is shown in Fig. 13. It can be seen that when there is a possible collision (Fig. 13(a)), the two robots will adjust their local trajectories respectively, as shown in Fig. 13(b). When there is no collision risk, the robot returns to the global trajectory tracking path planned by the virtual structure method, as shown in Fig. 13(c). However, there are some idealizations in the simulation, such as the lack of ocean current disturbance and random environmental disturbance(environmental noise). Therefore, we further carried out the pool experiments.

4.2. *Experimental results*

To verify the ability of the VO method in avoiding moving obstacles, the trajectory tracking and obstacle avoidance experiments of a single robot were carried out. The influence range (obstacle radius) of the obstacle is 15 cm which can ensure that the collision between the robot and the obstacle is avoided.

The blue pentagram and the blue diamond in Fig. 14 represent the starting point and the ending point, respectively, whose positions are shown in Table 3. The obstacle track is a straight yellow line, and the obstacle avoidance threshold is 0.7 m, that is, the obstacle avoidance strategy is started when the distance between the robot and the obstacle is 0.7 m. The robot starts tracking the desired trajectory from Fig. 14(a). When approaching the obstacle, the robot begins to avoid the obstacle, as shown in Fig. 14(b). After successfully avoiding the obstacle, the robot continues to track the desired trajectory, as shown in Fig. 14(c). In Fig. 14(d), the robot reaches the end of the desired trajectory. The effect of the proposed algorithm satisfies both the control input constraint and moving obstacle avoidance constraint. Fig. 15 shows the distance between the robot and the obstacle. The 49.4 cm marked point on the figure is the minimum distance, which is larger than the addition of robot radius (30.0 cm) and obstacle radius (15.0 cm). That is, the minimum distance between the robot and the obstacle is 4.4 cm. To avoid the instability during the propeller suddenly starting under water and the performance instability of the thruster at low speed or high speed,

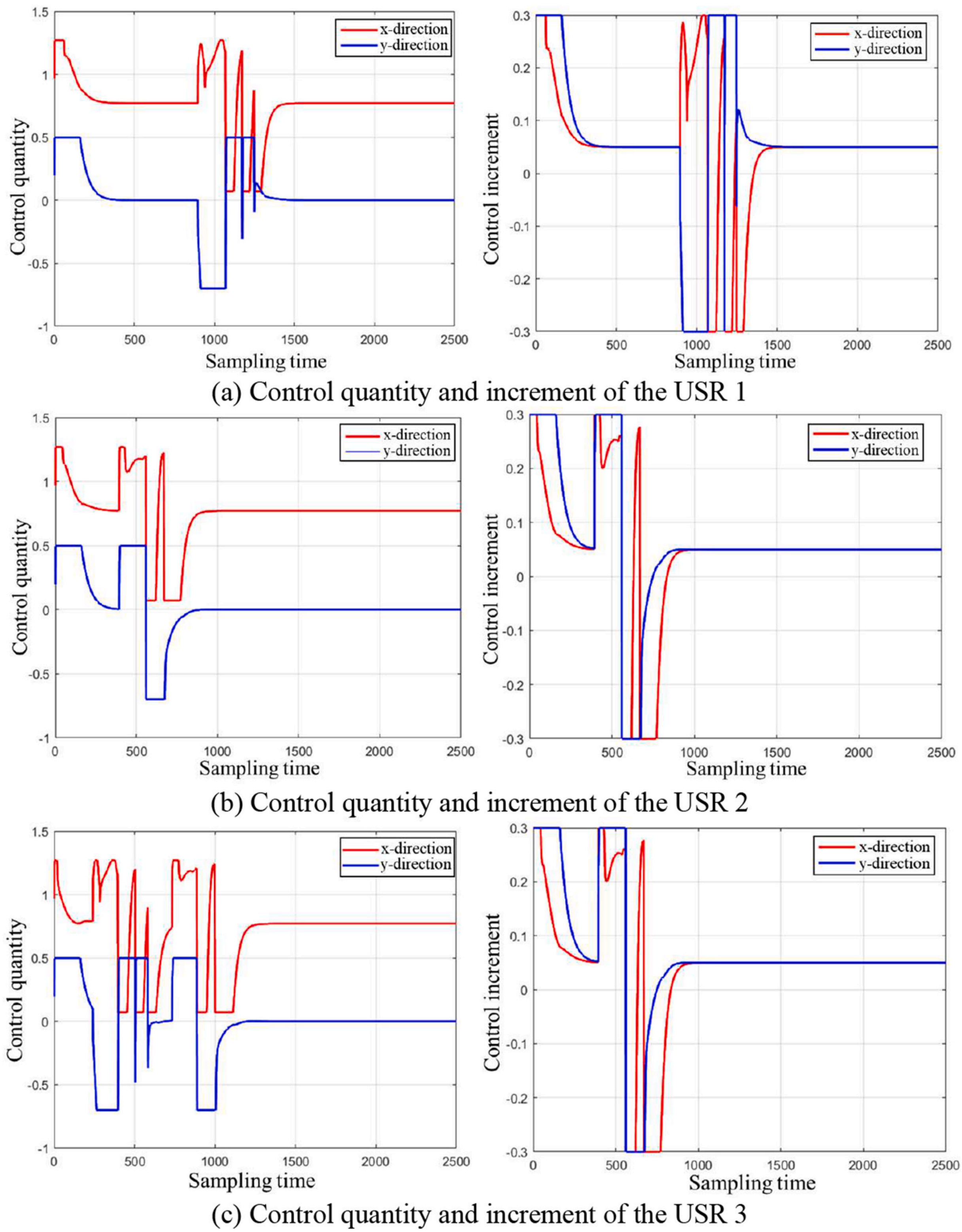

Fig. 19. The control quantity and increment of the USR in the formation obstacle avoidance experiment.

the output thrust base value of the propeller is set as 1.7 N. The robot is stationary under water when all four propellers reach base value. Then, by adjusting the input control quantity of each propeller based on the reference value, the movement of the robot is realized. Fig. 16 shows the robot's X and Y axis control inputs. The solid green line represents the base value of 1.7 N, and the control input constraint is 0.4 N. The algorithm satisfy the control constraint.

Then, the formation control and obstacle avoidance effects were verified. In Fig. 17, the start and the end points of the yellow robot are represented by the blue pentagram and diamond, respectively, whose positions are shown in Table 4. The starting and end points of the red robot are green pentagram and green diamond, respectively. The obstacle track is the yellow line. To make the obstacle affect two robots at the same time, the obstacle radius is set to 0.15 m, and the obstacle

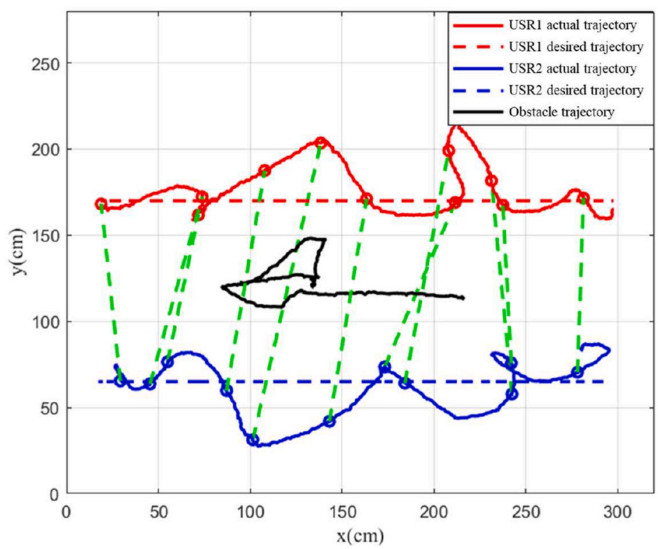

Fig. 20. Obstacle avoidance of formation members and obstacle trajectory diagram.

avoidance threshold is set to 0.90 m. When the motion begins, the robot tracks the desired trajectory, as shown in Fig. 17(a). When encountering obstacles, the robot formation avoids obstacles, as shown in Fig. 17(b). In Fig. 17(c), the robot avoids an obstacle chasing from behind the robot after returning to the desired trajectory. In Fig. 17(d), the obstacle leaves the obstacle avoidance range of the robot, and the robots retrack the expected trajectory to reach the end point of the expected trajectory. In the experiment, the robots maintained formation and avoided moving obstacles twice before continuing to track the desired trajectory. As a result, the tracking and obstacle avoidance of the robots are realized in the experiment. However, compared with the simulation results, the fluctuation in the experiment is larger, which is due to the small space in the pool experiment and the reflection of the water flow fluctuation through the wall caused by the robot movement, forming a strong interference flow field environment.

Fig. 18 shows the distance between robots and the obstacle in the comprehensive formation experiment. The minimum distances between the two robots and the obstacle are 69.6 cm and 73.5 cm, respectively. Both are larger than the radius sum 45.0 cm of the robot and the obstacle.

5. Discussion

5.1. The control quantity and increment

In simulation, the variation of the control quantity and increment of each robot is shown in Fig. 19. It can be seen that the y-direction control of each robot is between −0.7N∼0.5N, the x-direction control is between 0.1N∼1.3N, and the control increment is between −0.3N∼0.3N. The function of adaptive linear model predictive control for the processing of control quantity and increment constraints is verified.

5.2. Analysis of pool experiments

In pool experiment environment, the red line in Fig. 20 indicates that the trajectory of USR1 is located in the positive direction of the y-axis of the obstacle, where the solid line is the actual trajectory and the dashed line represents the desired trajectory without considering obstacle avoidance. The blue line indicates that the USR2 trajectory is located in the negative y-axis direction of the obstacle, where the solid line represents the actual trajectory and the dashed line represents the desired trajectory without obstacle avoidance. The solid black line in the middle represents the obstacle trajectory. The green dotted line connecting the two robots represents the actual formation at the same time, and it can be seen that the proposed algorithm can meet the expected trajectory of the formation and recover the formation after obstacle avoidance.

The current disturbances were not added in the simulation, to clearly reflect the performance of the strategy. However, in the experiment, due to the robot movement and the wall reflection of the experimental pool on the water flow, large random flows are formed, which are large disturbances for the robot. The robot deals with underwater current and noise disturbance in real time by AMPC and error feedback caused by disturbance.

Then, the obstacle rate change is analyzed, which varies in the range of about 6 cm/s∼16 cm/s, and the expected rate of the robot is 10 cm/s. In order to allow the obstacle to affect the two robots at the same time, the obstacle avoidance threshold is 0.9m, and the obstacle radius is set to 15 cm, which is also the reason for the large fluctuation of the formation trajectory error, but the overall effect is ideal.

In addition, the expected time of formation trajectory tracking is analyzed, and the expected time without obstacle avoidance behavior is about 28s, and the actual time is 32.66s, and the obstacle avoidance time only increases by 16.64 %, which also verifies the performance of the proposed obstacle avoidance strategy.

5.3. Discussion about limitation

It can be seen that in the experiment, under the condition of more disturbance and uncertainty than in the simulation, the underwater robot formation obstacle avoidance can be realized. However, the motion control stability of the underwater robot in the experiment is relatively poor than in the simulation. Therefore, if robots are actually used in the shallow sea environment, the anti-interference ability of our algorithm needs to be enhanced. This research mainly focuses on the successful formation obstacle avoidance of robots. The stronger anti-interference ability of robots and the fault-tolerant ability of communication are also problems that we need to focus on in the future.

The pool experimental environment has limited width and is less complex than real underwater exploration environment scenarios. The water flow disturbance in the pool is mostly a small amplitude random disturbance, which is formed by the changes of water flow and reflection of pool wall during the robot movement, and lacks the larger amplitude ocean current disturbance in the real application environment. In pool experiments, fibre-based communication is chosen to reduce communication delays or failures. In the actual environment, underwater acoustic communication will increase the communication cost and errors, which needs further research. In addition, in experiments, the upper-level decision-making are processed through the upper computer. In the actual environment, the processor needs to be concentrated on the robot, which requires more computing resources and energy. Therefore, there are some limitations in the pool experiment, which can be used as the preliminary verification of the improved algorithm.

6. Conclusion

In this paper, a moving obstacle avoidance constrained adaptive model predictive controller (MOAC-AMPC) is designed. Firstly, the hydrodynamic model is linearized. Then, an adaptive weight matrix based on tracking error is designed to reduce the tedious adjustment of the weight matrix. In addition, the adaptive weight matrix can optimize the tracking by reduce the tracking time. Then, to avoid multiple moving obstacles during tracking, the VO method is used to design the velocity constraint, so that USR can track the target trajectory without replanning the new trajectory. At the same time, a new constraint transformation method is proposed, which transforms the velocity constraint into the control increment constraint. Finally, the effectiveness of the algorithm is verified by numerical simulation and experiments in pool environment. Simulation and experimental results show that USR can effectively avoid multiple moving obstacles and avoid other formation members in formation in relatively low disturbance scenarios such as pool experiments. In the future, we will further study relevant adaptive control strategies and expand the application range in more realistic environments, such as considering more complex disturbance environments, obstacle environments, and solving communication delays and failures, to illustrate the reliability of our method and make further improvements.

CRediT authorship contribution statement

Ao Li: Software, Methodology, Conceptualization. **Shuxiang Guo:** Writing – review & editing, Supervision, Resources, Funding acquisition. **Chunying Li:** Writing – review & editing, Writing – original draft, Resources, Funding acquisition. **He Yin:** Methodology. **Meng Liu:** Software, Investigation. **Liwei Shi:** Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported in part by the Shenzhen Science and Technology Program under Grant RCBS20231211090725048, Shenzhen, China, in part by the GuangDong Basic and Applied Basic Research Foundation under Grant 2025A1515011007, Guangdong, China, in part by the High level of special funds under Grant G03034K003 from Southern University of Science and Technology, Shenzhen, China, in part by the Shenzhen "Medical Engineering Integration" Special Support Program Project under Grant F-2024-Z99-502365, Shenzhen, China, in part by the National High-Tech Research and Development Program (863 Program) of China under Grant 2015AA043202, in part by the National Natural Science Foundation of China under Grant 62273042.

Appendices A. Velocity Estimation Based on Kalman Filter

To enhance the recognition accuracy of the velocity of the USR and the obstacle, considering the interference factors such as delay, the Kalman filter method was used to obtain the velocity of USR in the nonlinear state. The state transition and the observation equations of the Kalman filter can be expressed as:

$$\begin{cases} \boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}) + \varepsilon_k \\ \boldsymbol{z}_k = h(\boldsymbol{x}_k) + \delta_k \end{cases} \tag{A.1}$$

where $\boldsymbol{x}_k$ is the state quantity, $\boldsymbol{z}_k$ is the observational measurement, ε_k is the process noise, and δ_k is the observation noise.

The prediction of the state process first deduces the state transition equation according to the dynamic equation of the USR model, satisfy:

$$\begin{bmatrix} \widehat{x_k} \\ \widehat{y_k} \\ \widehat{v_k} \\ \widehat{\theta_k} \end{bmatrix} = \begin{bmatrix} x_{k-1} + v_{k-1} \cos \theta_{k-1} \Delta t \\ y_{k-1} + v_{k-1} \sin \theta_{k-1} \Delta t \\ v_{k-1} \\ \theta_{k-1} \end{bmatrix} \tag{A.2}$$

where $\widehat{x_k}$, $\widehat{y_k}$, $\widehat{v_k}, \widehat{\theta_k}$ in the state vector represent the predicted values of the horizontal and vertical position, velocity, and heading angle of the robot in the earth coordinate system, respectively. The covariance matrix for state estimation is $P_k^- = F_{k-1} p_{k-1} F_{k-1}^T + Q.P_k^-$. P_k^- is iteratively obtained by setting the initial covariance matrix. F_{k-1} is a Jacobian matrix resulting from the linearization of x by a nonlinear system.

The adjustment update process can be expressed as:

$$G_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + R \right)^{-1} \tag{A.3}$$

And then the optimal value of the pose state can be estimated, satisfy $\widehat{x_k} = \widehat{x_k} + G_k(z_k - h(\widehat{x_k}))$. Where $z_k = [x_k, y_k, \theta_k]^T$. H_k is the partial derivative of $h(x)$.

References

An, R., Guo, S., Yu, Y., Li, C., Awa, T., 2022a. Task planning and collaboration of jellyfish-inspired multiple spherical underwater robots. J. Bionic Eng. 19 (3), 643–656.

An, R., Guo, S., Liang, Z., 2022b. Uncertain moving obstacles avoiding method in 3d arbitrary path planning for a spherical underwater robot. Robot. Autonom. Syst. 151, 104011.

Ding, W., Zhang, L., Zhang, G., Wang, C., Chai, Y., Yang, T., Mao, Z., 2024. Research on obstacle avoidance of multi-AUV cluster formation based on virtual structure and artificial potential field method. Comput. Electr. Eng. 117, 109250.

Erskine, J., Balderas-Hill, R., Fantoni, I., Chriette, A., 2021. Model predictive control for dynamic quadrotor bearing formations. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 124–130. Xi'an, China, Jun.

Fu, X., Pan, J., Wang, H., Gao, X., 2020. A formation maintenance and reconstruction method of uav swarm based on distributed control. Aero. Sci. Technol. 104, 105981.

Han, G., Qi, X., Peng, Y., Lin, C., Zhang, Y., Lu, Q., 2022. Early warning obstacle avoidance-enabled path planning for multi-auv-based maritime transportation systems. IEEE Trans. Intell. Transport. Syst. 24 (12), 2656–2667.

Heshmati-Alamdari, S., Bechlioulis, C.P., Karras, G.C., Kyriakopoulos, K.J., 2021. Cooperative impedance control for multiple underwater vehicle manipulator systems under lean communication. IEEE J. Ocean. Eng. 46 (2), 447–465.

Hou, X., Xing, H., Guo, S., Shi, H., Yuan, N., 2024. Design and implementation of a model predictive formation tracking control system for underwater multiple small spherical robots. Appl. Sci. 14 (1), 294.

Li, C., Guo, S., 2022. Performance evaluation of spherical underwater robot with attitude controller. Ocean Eng. 268, 113434.

Li, C., Guo, S., 2023. Adaptive multi-mode switching strategy for the spherical underwater robot with hybrid thrusters. Adv. Eng. Inform. 55, 101845.

Li, A., Guo, S., 2024. Surface hydrodynamic characteristic analysis of underwater robots applied to perception of lateral lines. Phys. Fluids 36 (3).

Li, A., Guo, S., Liu, M., Yin, H., 2022a. Hydrodynamic characteristics-based adaptive model predictive control for the spherical underwater robot under ocean current disturbance. Machines 10 (9), 798.

Li, C., Guo, S., Guo, J., 2022b. Study on obstacle avoidance strategy using multiple ultrasonic sensors for spherical underwater robots. IEEE Sens. J. 22 (24), 24458–24470.
Li, A., Guo, S., Li, C., 2024. An artificial lateral lines-based an active obstacle recognition strategy and performance evaluation for bionic underwater robots. IEEE Sens. J. 24 (16), 26266–26277.
Liu, C.Z., Li, L., Yong, J.W., Muhammad, F., Cheng, S., Wang, X.Y., Li, W.B., 2021. The bionics and its application in energy management strategy of plug-in hybrid electric vehicle formation. IEEE Trans. Intell. Transport. Syst. 22 (12), 7860–7874.
Meng, C.C., Zhang, W., Du, X., 2023. Finite-time extended state observer based collision-free leaderless formation control of multiple AUVs via event-triggered control. Ocean Eng. 268 (Jan).
Pan, Z., Sun, Z., Deng, H., Li, D., 2022. A multilayer graph for multiagent formation and trajectory tracking control based on mpc algorithm. IEEE Trans. Cybern. 52 (12), 13586–13597.
Pang, W., Zhu, D.Q., Sun, C.Y., 2024. Multi-AUV formation reconfiguration obstacle avoidance algorithm based on affine transformation and improved artificial potential field under ocean currents disturbance. IEEE Trans. Autom. Sci. Eng. 21 (2).
Peng, Y.D., Guo, L.C.A., Meng, Q.H., Chen, H.Q., 2022. Research on hover control of AUV uncertain stochastic nonlinear system based on constructive backstepping control strategy. IEEE Access 10, 50914–50924.
Peng, Z.H., Cui, K., Li, H.J., Gu, N., Liu, L., Wang, D., 2024. Model-free anti-disturbance autopilot design for autonomous surface vehicles with hardware-in-loop experiments. IEEE Trans. Ind. Inf. 20 (2), 2387–2396.
Shi, L., Hu, Y., Su, S., Guo, S., Xing, H., Hou, X., Liu, Y., Chen, Z., Li, Z., Xia, D., 2020. A fuzzy pid algorithm for a novel miniature spherical robots with three-dimensional underwater motion control. J. Bionic Eng. 17 (5), 959–969.
Tang, J.G., Chang, J., Li, B., Zhang, A.Q., 2021. Modeling and control of hybrid-driven gliding motion for an underwater gliding snake-like robot. Int. J. Control Autom. Syst. 19 (9), 3190–3198.
Wang, W.R., Yan, J.H., Wang, H., Ge, H.L., Zhu, Z.Y., Yang, G.J., 2022. Adaptive MPC trajectory tracking for AUV based on Laguerre function. Ocean Eng. 261, 111870.
Wang, K.H., Zou, W., Ma, R.C., Wang, Y., Su, H., 2023. Model predictive trajectory tracking control of an underactuated bionic underwater vehicle. IEEE ASME Trans. Mechatron. 29 (3), 1690–1701.
Wei, H., Shen, C., Shi, Y., 2021. Distributed lyapunov-based model predictive formation tracking control for autonomous underwater vehicles subject to disturbances. IEEE Trans. Syst. Man Cybern. -Syst. 51 (8), 5198–5208.
Wen, J.B., Yang, J.C., Li, Y., He, J.Y., Li, Z.J., Song, H.B., 2023. Behavior-based formation control digital twin for multi-AUG in edge computing. IEEE Trans. Netw. Sci. Eng. 10 (5), 2791–2801.
Xing, H., Shi, L., Hou, X., Liu, Y., Guo, S., 2021. Design, modeling and control of a miniature bio-inspired amphibious spherical robot. Mechatronics 77, 102574.
Xing, H., Chang, L., Guo, S., Shi, L., Liu, W., Li, H., Zhao, Y., 2022. Research on multi-mode motion of the miniature amphibious turtle-inspired robot. Robot 44 (2), 212–223.
Yin, H., Guo, S., Liu, M., 2022. A virtual linkage-based dual event triggered formation control strategy for multiple amphibious spherical robots in constrained space with limited communication. IEEE Sens. J. 22 (13), 13395–13406.
Yin, H., Guo, S., Li, A., Shi, L., Liu, M., 2023. A deep reinforcement learning-based decentralized hierarchical motion control strategy for multiple amphibious spherical robot systems with tilting thrusters. IEEE Sens. J. 24 (1), 769–779.
Zhao, Y., Lu, W.L., Chen, Y.Q., 2024. Switching dynamic event-triggered prescribed performance control for underactuated ASVs. IEEE Trans. Veh. Technol. 73 (10), 14370–14381.
Zhen, Q.Z., Wan, L., Li, Y.L., Jiang, D.P., 2022. Formation control of a multi-AUVs system based on virtual structure and artificial potential field on SE (3). Ocean Eng. 253, 111148.
Zhen, Q., Wan, L., Zhang, Y., Jiang, D., 2024. Consensus-based formation control and gyroscopic obstacle avoidance for multiple autonomous underwater vehicles on SE (3). J. Mar. Sci. Eng. 12 (12), 2350.