Proceedings of 2022 IEEE
International Conference on Mechatronics and Automation
August 7 - 10, Guilin, China

Dynamic Obstacles Avoiding Method of Formation Tracking Control for Amphibious Spherical Robots

Meng Liu, Shuxiang Guo*, He Yin, Ao Li, Mugen Zhou
Key Laboratory of Convergence Medical Engineering System and Healthcare Technology School of Life Science, Beijing Institute of Technology
No.5, Zhongguancun South Street, Haidian District, Beijing 100081, China
E-mails: liumeng@bit.edu.cn; yinhe@bit.edu.cn; liao@bit.edu.cn
**Corresponding author: guoshuxiang@bit.edu.cn*

***Abstract*—The collision-free formation trajectory tracking method is established for multiple amphibious spherical robots (multi-ASRs) in this paper. This approach combine the Velocity Obstacle (VO) and Model Predictive Controller (MPC) to achieve obstacle avoidance during the formation tracking. The kinematics and dynamics constrains ignored by VO are considered by MPC. The collsion-free velocity constraints for MPC is generated by VO. First, the current velocity of obstacle is used to generate the safe zones for the new velocity of ASR. Then, the safe zones are used to generate constraints for output variables. Finally, the optimization problem which contain formation keeping conditions is solved. Simulations are carried out to validate the feasibility of VO-MPC method.**

Index terms— Amphibious spherical robot, Dynamic obstacles avoiding, Formation tracking, Model predictive control, Velocity obstacle

I. Introduction

With the increase in number of robots, multi-ASR systems with strong robustness can effectively perform large-scale search and exploration missions. Obstacle avoidance is a critical problem for multi-robot to work safely in complex environment. Compared with static obstacles, dynamic obstacles have stronger randomness and uncertainty. In the last decade, the research of multiple autonomous underwater vehicle(AUV) [1–3] has attracted more attention from engineers and scientists. Formation trajectory tracking is core problem of multi-AUV system. Every AUV is required to avoid collision with other formation members when tracking a given reference trajectory.

Most of formation control methods of AUVs can be divided into two categories: leader-follower and virtual structure. However, for AUV formation control, the above method cannot satisfy constraints in various practical environments (such as the saturation of output [4] and its increment per unit time) on the premise of optimal control performance. Compared with the above methods, model predictive control (MPC) can deal with the system constraints brought by the actual environment under the premise of optimizing the system control effect. Many researchers [5–9] use MPC to achieve trajectory tracking or formation control. A model predictive controller based on distributed Lyapunov[6] is designed to solve the formation control problem of AUV. This study considers the interference of ocean current and collision avoidance to other team members. A decentralized controller based on MPC [7] considers the constraints of attitude kinematics in formation control, improves the existing rigidity-based and visual servoing method that only uses quadrotor as integrator, and fully optimizes the high maneuverability of quadrotor. A Multilayer Graph [8] is used to track the desired trajectory of the virtual leader while the robot maintains desired angle and distance. A fish swarm optimization algorithm (FSO) [9], which simulates the predation behavior of fish, is used to solve the nonlinear MPC problem and generate the optimal pilot in top layer. A decentralized intelligent cruise method is used in the middle layer to make followers track leader by mimicing behavior of fish.

To ensure the safety of formation members, collision avoidance must be considered in the disign process of multi-AUV formation controller. Several methods [10–12] have been proposed for this problem. The Lyapunov barrier function [10] is used to describe collision avoidance in the gradient-dependent control solution. Collision avoidance conditions are modeled as constraints [11] in the quadratic optimization problem. The additional potential field term [11] based on the distance of formation members is introduced to avoid collisions in a virtual linkage-based formation control strategy.VO method is a local obstacle avoidance and navigation algorithm that enables the robot to move at non-collision velocity between several obstacles. Velocity obstacle cone is the robot velocity collection of future collision with obstacles. The basic principle of velocity obstacle method is to avoid the velocity obstacle cone under the condition that robot velocity is as close to desired velocity as possible. VO has been gradually developed into Reciprocal Velocity Obstacle (RVO) and Optimal Reciprocal Collision Avoidance (ORCA) in subsequent reserches. Many studies [13–17] have applied VO to multi-robot navigation. A distributed multi-robot obstacle avoidance method [16] using deep reinforcement learning with RVO is proposed to solve the problem of collision avoidance problems with limited information. To avoid the ship in violation of COLREG rules, probabilistic velocity obstacle algorithm [17] was designed to achieve robust collision avoidance based on the fusion of intent reasoning and velocity obstacle. It balances compliance with traffic rules and active avoidance behaviors.

978-1-6654-0853-0/22/$31.00 ©2022 IEEE

The MPC method can calculate the control input in the future time period under the system dynamics equation and various inequality constraints. Reasonable design of cost functions and constraints in this method make the robot track desired trajectory and avoid obstacles. Some properties of VO are easy to apply in convex optimization problems. Therefore, several studies combine MPC and VO for obstacle avoidance of single or multiple robots. ORCA is combined with MPC [18] based on flatness to generate local collision free trajectory for each quadrotor aircraft. It improves the smoothness of the generated trajectory and improves the safety of high-velocity maneuvering mode. In each running step of algorithm[19] based on ORCA, the desired trajectory, reference control input and collision-avoiding velocity cone are generated according to the new velocity. The collision-avoiding velocity cone is converted into the robot velocity constraint in MPC. Inspired by [19], a collsion-free formation control strategy is established in this paper.

Inspired by above approaches, a collsion-free formation control strategy is established in this paper to finish trajectory tracking and collsion avoidance for multi-ASR.

Comparing the traditional VO, the kinematics and dynamics constrains are considered by VO-MPC. The collsion-free velocity constraints for MPC is generated by VO. First, A distributed MPC formation tracking method is established. The control vector of each AUV is calculated by solving the online optimization problem. Secondly, velocity constrain for ASR during the prediction horizon is generated by VO method. To make ASR avoid the collsion to obstacle, this constrain is introduced into the optimization problem. Finally, we conduct two sets of simulation to validate the feasibility of method.

The rest of this paper is organized as follows. The ASR modeling and problem formulation is established in section II. The VO-MPC formation tracking algorithm is designed in Section III. Some simulation results are shown in Section IV and conclusion of this paper is shown in Section V.

Notations: The $\|\boldsymbol{x}\|$ denotes 2-norm. The Euclidean norm with positive weight matrix $\boldsymbol{\Psi}$ is deboted by $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Psi}$ and $\left[\boldsymbol{x}_1^{\mathrm{T}} \ldots \boldsymbol{x}_n^{\mathrm{T}}\right]^{\mathrm{T}}$ denotes the column vector $\left[\boldsymbol{x}_1^{\mathrm{T}} \ldots \boldsymbol{x}_n^{\mathrm{T}}\right]^{\mathrm{T}}$. The superscript "T" denotes the transposition operation. The diag$(\cdot)$ represents the diagonal matrix.

II. Problem Formulation

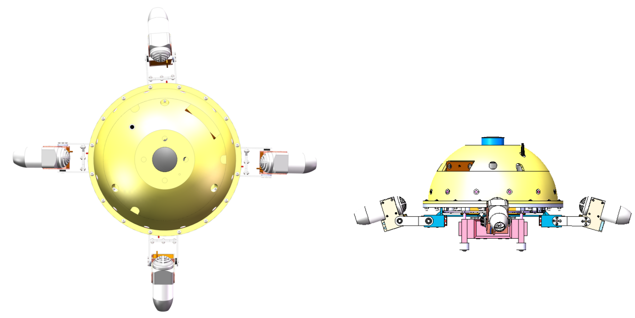
A. ASR Modeling

This study only consider the horizontal motion of the ASR [20]. We assumed that the pitch angle and roll angle are small and neglect the corresponding elements such as roll, heave and pitch. The mechanical structure of ASR is shown in Fig. 1.

We consider the ASRs team with same dynamics. The kinematics of the ith ASR can be shown as follows

$$\dot{\boldsymbol{\eta}}_i = \boldsymbol{v}_i \tag{1}$$

where $\boldsymbol{\eta_i} = [x_i, y_i]^{\mathrm{T}}$ is position vector, and $\boldsymbol{v}_i = [u_i, v_i]^{\mathrm{T}}$ denotes velocity vector. The 3-DOF dynamic motion formula



(a) Top view (b) Side view

Fig. 1. The ASR in "X" motion mode.

of ASR i can be described as follows

$$\boldsymbol{M}_i\dot{\boldsymbol{v}}_i + \boldsymbol{C}_i(\boldsymbol{v}_i)\boldsymbol{v}_i + \boldsymbol{D}_i(\boldsymbol{v}_i)\boldsymbol{v}_i + \boldsymbol{g}_i(\boldsymbol{\eta}_i) = \boldsymbol{\tau}_i \tag{2}$$

where $\boldsymbol{M_i}$ is inertia matrix including added mass of ASR i. Damping matrix is described as $\boldsymbol{D}_i(\boldsymbol{v}_i)$. Assuming that gravity and buoyancy are canceled by each other, the restoring force $\boldsymbol{g}_i(\boldsymbol{\eta}_i) = \mathbf{0}$. $\boldsymbol{\tau}_i$ represents the control vector, and $\boldsymbol{C}_i(\boldsymbol{v}_i)$ represents the Coriolis-centripetal matrix. System state model for formation tracking problem of ASR i is established by combining (1) and (2),

$$\begin{aligned} \dot{\boldsymbol{x}}_i &= \begin{bmatrix} \boldsymbol{v}_i \\ \boldsymbol{M}_i^{-1}\left(\boldsymbol{\tau}_i - \boldsymbol{C}_i(\boldsymbol{v}_i)\boldsymbol{v}_i - \boldsymbol{D}_i(\boldsymbol{v}_i)\boldsymbol{v}_i\right) \end{bmatrix} \\ &= f(\boldsymbol{x}_i, \boldsymbol{\tau}_i) \end{aligned} \tag{3}$$

where $\boldsymbol{x}_i = \mathrm{col}(\boldsymbol{\eta}_i, \boldsymbol{v}_i) \in \mathbb{R}^6$ is the state vector and $\boldsymbol{\tau}_i \in \mathbb{R}^3$ is the control input vector of ASR i.

B. Problem Formulation

In order to describe information communication between ASRs better, a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is introduced, where $\mathcal{V} = \{1, \ldots, N\}$ is the set of nodes which denote the ASRs and the node i in graph represents ASR i. $\mathcal{E} = \{(i,j) \subset \mathcal{V} \times \mathcal{V}\}$ is the set of all edges related with element (i,j) . The edges set describes the information exchange from node i to node j. Associated with the graph $\mathcal{G}$, the adjacency matrix $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{N\times N}$ is defined such that $a_{ij} = 1$ if $(i,j) \in \mathcal{E}$ which denotes the information exchanged between ASR i and j is available, and $a_{ij} = 0$ otherwise. We consider that communication topology following the assumption.

Assumption 1: Each ASR i in multi-ASR system can receive information of its neighbors and the virtual leader. Each robot is synchronized on the time clock and the system state is valid during each sampling period. We established model of the formation tracking and collision avoidance for each ASR. In order to maintain the pre-set expectation formation shape and track the time-varying expectation trajectory of the virtual leader $\boldsymbol{\eta}_r(t)$, ASR i need satisfy:
1) Tracking reference: $\lim_{t\to\infty}\{\boldsymbol{p}_i(t) - \boldsymbol{p}_r(t)\} = \boldsymbol{d}_{ir}$
2) Formation control: $\lim_{t\to\infty}\{\boldsymbol{p}_i(t) - \boldsymbol{p}_j(t)\} = \boldsymbol{d_{ij}}$
where $\boldsymbol{d}_{ir}$ is the configuration vector of formation shape for

each ASR. $\boldsymbol{d}_{ij} = \boldsymbol{p}_i - \boldsymbol{p}_j$ denotes the relative position vector between ASR i and ASR $j, i \neq j$. $\boldsymbol{p}_r = [x_r, y_r]^{\mathrm{T}}$ denotes the time-varying desired position vector of virtual leader.

III. Formation Tracking Controller with Dynamic Obstacle Avoiding

A. Disign of Formation Controller Based on MPC

To meet control requirement for each ASR $i, i \in \mathcal{V}$, the cost function at time instant t_k can be defined as

$$J_i = \int_{t_k}^{t_k+T} (J_{ij}^{fo} + J_i^{\tau}) ds \quad s \in [t_k, t_k + T] \tag{4}$$

where T denotes the prediction horizon, t_s denotes the sampling period and the relationship of t_k and t_s is $t_{k+1} = t_k + t_s$. J_{ij}^{fo} denotes the fromation tracking error and J_i^{τ} denotes the energy consumption of ASRs

In order to ensure that the formation members track the desired trajectory and keep the formation shape in the future, we design the formation keeping term in the cost function by using the tracking errors and relative positions.

$$J_{ij}^{fo} = \sum_{j=1}^{N} \left[a_{ij} \left\| \boldsymbol{x}_{ij}(s; t_k) \right\|_{\boldsymbol{Q}_i}^2 + \right] \tag{5}$$

where $\boldsymbol{Q}_i \geq 0$ and $\boldsymbol{R}_i > 0$ are weight matrices. $\boldsymbol{x}_{ir}$ and $\boldsymbol{x}_{ij}$ are as follows

$$\boldsymbol{x}_{ir} = \tilde{\boldsymbol{x}}_i + \boldsymbol{\Lambda}_i - \boldsymbol{x}_r \tag{6}$$

where $\boldsymbol{x}_r = \operatorname{col}(\boldsymbol{\eta}_r, \boldsymbol{v}_r)$ denotes the desired state and $\boldsymbol{\Lambda}_i = \operatorname{col}(\boldsymbol{d}_{ir}, \boldsymbol{0})$. $\boldsymbol{x}_i$ is the kinematics state of ASR i, $\tilde{\boldsymbol{x}}_i$ represents the predicted trajectory in a period of future time, which is updated according to the state space representation. The $\tilde{\boldsymbol{x}}_i$ are generated by following equation.

$$\dot{\tilde{\boldsymbol{x}}}_i(s; t_k) = f(\tilde{\boldsymbol{x}}_i(s; t_k), \boldsymbol{\tau}_i(s; t_k)) \tag{7}$$

In order to minimize total energy consumption in the future, the control input in the predicted domain is used to design the energy consumption term.

$$J_i^{\tau} = \left\| \boldsymbol{\tau}_i(s; t_k) \right\|_{\boldsymbol{R}_i}^2 \tag{8}$$

Considering the initial values, state transition equations and control input maximum $\overline{\boldsymbol{\tau}}$ and minimum $\underline{\boldsymbol{\tau}}$, the optimization problem is established.

$$\begin{gathered} \underset{\tau_i}{\arg\min}\, J_i(\boldsymbol{x}_i, \boldsymbol{\tau}_i) \\ \dot{\tilde{\boldsymbol{x}}}_i(s; t_k) = f(\tilde{\boldsymbol{x}}_i(s; t_k), \boldsymbol{\tau}_i(s; t_k)) \\ \tilde{\boldsymbol{x}}_i(t_k; t_k) = \boldsymbol{x}_i(t_k) \\ \underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau}_k \leq \overline{\boldsymbol{\tau}} \end{gathered} \tag{9}$$

The primary task of this paper is to keep the ASRs moving towards their target in formation shape while avoiding collisions. This problem is formulated as a optimization problem (9) for every ASR. Then, how the VO algorithm provides velocity constraints for the MPC problem formulation is discussed.

B. Design of Velocity Constraint Based on VO

The VO-MPC algorithm established in this paper is a distributed formation collision avoidance algorithm, which allows each ASR to independently calculate the optimal velocity in the current step. Specifically, the algorithm assumes that each ASR can obtain the information containing relative distance and velocity of obstacle. Based on these information, the ASR calculates velocity obstacle cone, which is the set of velocities that may collide with obstacle within a future interval. Then, a velocity outside the velocity obstacle cone (usually at the boundary) and closest to the desired ASR velocity is selected as the new velocity for the next step. The velocity constraint designed in this paper is as follows.

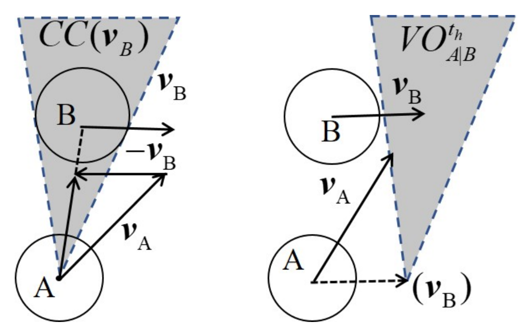
An open disc area of center $\boldsymbol{p}$ and radius r is defined as

$$D(\boldsymbol{p}, r) = \{\boldsymbol{q} \mid \|\boldsymbol{q} - \boldsymbol{p}\| < r\} \tag{10}$$

The $VO_{A|B}^{t_h}$ which means velocity obstacle of robot A to obstacle B during time horizon t_h is defined as

$$\begin{gathered} VO_{A|B}^{t_h} = \{\boldsymbol{v} \mid \exists t \in [0, t_h], \boldsymbol{v}t \in D_{rel}\} \\ D_{rel} = D(\boldsymbol{p}_B - \boldsymbol{p}_A, r_A + r_B) \end{gathered} \tag{11}$$

where $\boldsymbol{p}_A$ are the positions of robot A. $\boldsymbol{v}_A$ is current velocity of robot A. r_A is safe radius of robot A, which means there is a danger of collision if the distance to others is smaller than that. So the radius of safe circle is set larger than that of the ASR.



(a) Collsion cone $CC(v_B)$ (b) Velocity obstacle $VO_{AB}^{t_h}$

Fig. 2. Construction of the velocity obstacle cone.

Clearly, $VO_{A|B}^{t_h}$ denotes the set of relative velocities that there is a collision between the ASR and obstacle within time t_h. Note that for obstacle B, its velocity obstacle is $VO_{B|A}^{t_h} = -VO_{A|B}^{t_h}$. The collision cone $CC(v_B)$ shown in Fig. 2(a) can be converted into velocity obstacle $VO_{A|B}^{t_h}$ by relative velocity. To enable ASR to avoid obstacles at a safe distance, we extend $VO_{AB}^{t_h}$ shown in Fig. 2(b), the new velocity $\boldsymbol{v}_{t+1}$ needs to meet two conditions:

1) The change of current velocity (norm of $\boldsymbol{u}$) should be as small as possible.

2) Avoiding obstacles $\left(\boldsymbol{v}_{t+1} \notin VO_{A|B}^{t_h}\right)$.

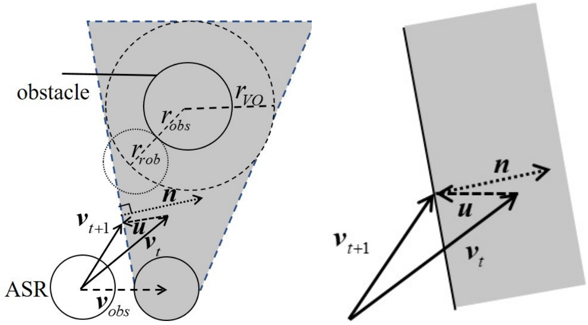
According to the above conditions, we extend the radius of the obstacle to $r_{VO} = r_{obs} + r_{rob}$ in Fig. 3(b), where r_{obs} is the radius of the obstacle, r_{rob} is the radius of ASR. The r_{VO}

is the basis for finding the velocity of obstacle avoidance.

$\boldsymbol{u}$ denotes the vector with minimum length that points from $(\boldsymbol{v}_A - \boldsymbol{v}_B)$ to the boundary of $VO^{\tau}_{A|B}$

$$\boldsymbol{u} = \underset{\boldsymbol{v} \in \partial VO^{\tau}_{A|B}}{\operatorname{argmin}} \ \|\boldsymbol{v} - (\boldsymbol{v}_A - \boldsymbol{v}_B)\| - (\boldsymbol{v}_A - \boldsymbol{v}_B) \qquad (12)$$

In other words, $\boldsymbol{u}$ is the smallest velocity change required so that the relevant velocity $\boldsymbol{v}_A - \boldsymbol{v}_B$ can "escape" from the velocity obstacle $VO^{th}_{A|B}$



(a) Extended velocity obstacle cone (b) velocity constraint

Fig. 3. Construction of the velocity constrain.

Finally, the set of optimal collision avoidance (OCA) velocity can be obtained as follows:

$$OCA^{\tau}_{ASR|obs} = \{\boldsymbol{v}_{t+1} \mid (\boldsymbol{v}_{t+1} - (\boldsymbol{v}_t + \boldsymbol{u})) \cdot \boldsymbol{n} \leq 0\} \qquad (13)$$

where n is the normal vector vertical with boundary of extended velocity obstacle cone. Convex constrained regions can be transformed into inequality constraints, and inequality constraints can also represent specific regions. The final result can be transformed into constraint as $\boldsymbol{v}_{t+1} \cdot \boldsymbol{n} \leq (\boldsymbol{v}_t + \boldsymbol{u}) \cdot \boldsymbol{n}$ shown in the Fig. 3(b).

The final optimization problem (14) can be obtained by substituting the velocity constraint of (13) into (9).

$$\begin{gathered} \underset{\tau_i}{\arg\min}\, J_i(\boldsymbol{x}_i, \boldsymbol{\tau}_i) \\ \dot{\tilde{\boldsymbol{x}}}_i(s; t_k) = f(\tilde{\boldsymbol{x}}_i(s; t_k), \boldsymbol{\tau}_i(s; t_k)) \\ \tilde{\boldsymbol{x}}_i(t_k; t_k) = \boldsymbol{x}_i(t_k) \\ \underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau}_k \leq \overline{\boldsymbol{\tau}} \\ OCA^{\tau}_{ASR|obs} \end{gathered} \qquad (14)$$

In the start of algorithm flow, the VO constraint of ASR can be calculated using the current position and velocity of ASR and obstacle. We use the current state of the ASR to predict its future position and velocity information which generates the VO constraint during predicting horizon N_p in the subsequent time steps $t+i \mid i = 1, 2 \ldots N_p - 1$. The position and velocity of ASRi are obtained from the state trajectory $\tilde{\boldsymbol{x}}$ predicted by (2). Instead, for obstacles, we use a simple motion model $\boldsymbol{p}_{obs}(t + T) = \boldsymbol{p}_{obs}(t) + \boldsymbol{v}_{obs}(t)t$ to predict their current positions. We assume that the velocity of the obstacle remains constant during the prediction horizon. The distributed VO-MPC formation tracking algorithm will be implemented in the predicted horizon for each ASR and the algorithm flow is shown in Algorithm 1.

Algorithm 1 VO-MPC Implementation.

1: **Initialization**:Set up the weight matrixs $\boldsymbol{Q}_i, \boldsymbol{Q}'_i, \boldsymbol{R}_i$, and other parameters for every ASR. Set time $k = 0$;
2: **procedure**
3: Sample the state varable of ASR i $\boldsymbol{x}_i(t_k)$;
4: Generate the velocity constraint (13)
5: ASR i solves (14), generating control sequence $\boldsymbol{\tau}_i(s; t_k), s \in [t_k, t_k + T]$
6: Apply the first term in control sequence $\boldsymbol{\tau}_i(s; t_k), s \in [t_k, t_k + T]$;
7: $k = k + 1$, go to step 3.
8: **end procedure**

IV. Simulation and Results

Two simulation scenarios are set to verify the feasibility of VO-MPC algorithm for multi-ASR formation. Formation tracking mission is tested in the first scenario. Obstacle avoidance by adding two moving obstacle in the environment is tested in the second scenario. Formation trajectory tracking performance and moving obstacles avoidance of the VO-MPC method are validate by simulation results.

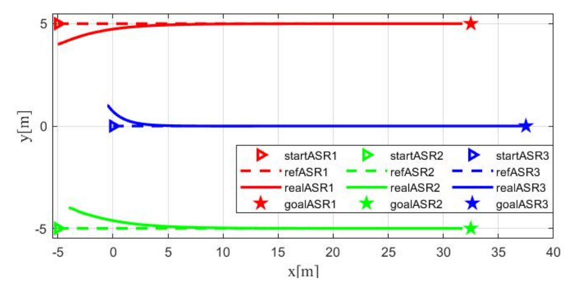


Fig. 4. Trajectory of ASRs and reference trajectory.

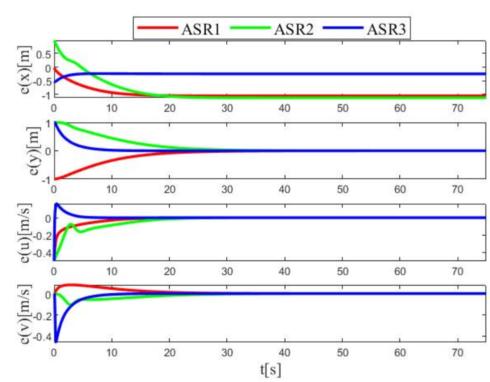


Fig. 5. Formation tracking errors of ASRs.

The initial state variable for the three ASRs in Fig. 4 are set as $[-5, 4]^{\mathrm{T}}$, $[0, 1]^{\mathrm{T}}$ and $[-4, 4]^{\mathrm{T}}$. The formation tracking problems of ASRs is considered without the dynamic obstacles avoidance in the first part. As shown in Fig. 4 and Fig. 5, every

ASR can track its respective desired trajectory with the desired formation shape. It can be seen from Fig. 6 that the control input component is limited to range between the minimum -4.2 N and maximum 4.2 N.

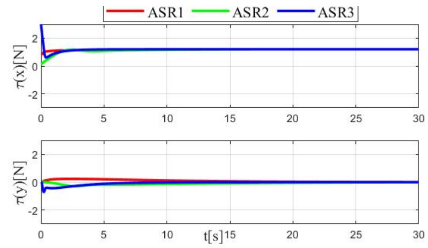
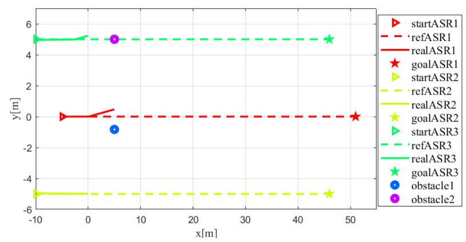
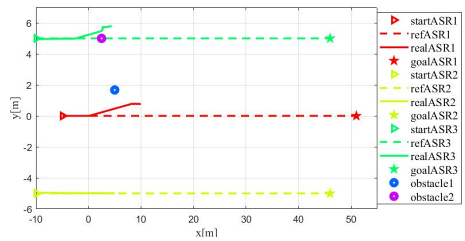


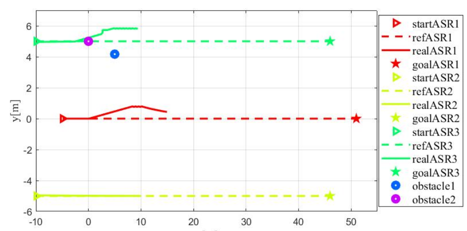
Fig. 6. Control input component of ASRs.



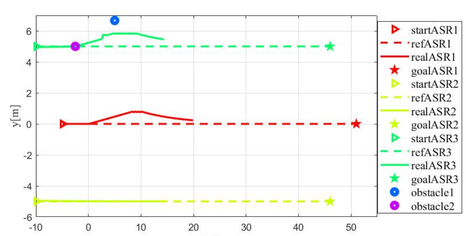
(a) t=20s



(b) t=30s



(c) t=40s



(d) t=50s

Fig. 7. Formation tracking with moving obstacle avoidance.

The formation tracking with moving obstacle avoidance is shown in Fig. 7. The velocity of obstacle1 is $[-0.25, 0]^{\mathrm{T}}$ m/s and that of the obstacle2 is $[0, 0.25]^{\mathrm{T}}$ m/s. ASR1 starts to avoid the obstacle1 and ASR2 starts to avoid the obstacle2 in Fig. 7(a). The trajectory of ASR3 in Fig. 7(b) changes rapidly because the distance to obstacle2 is small. The ASR3 is affected by obstacle1 after completing the avoidance of the obstacle2 in Fig. 7(c). Without the affect of obstacle1, the tracking error of ASR3 starts to decrease in Fig. 7(d).

Distance between obstacles and ASRs is shown in Fig. 8. dis_1 denotes the distance between ASR and obstacle1 and dis_2 represents the distance between ASR and obstacle2. Every ASR in formation avoid all moving obstacles. Tracking errors for every ASR shown in Fig. 9 validate the effectiveness of the cost function in (4). Control input component is shown in Fig. 10.

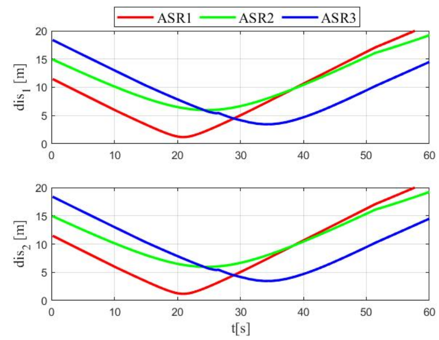


Fig. 8. Distance between obstacles and ASRs.

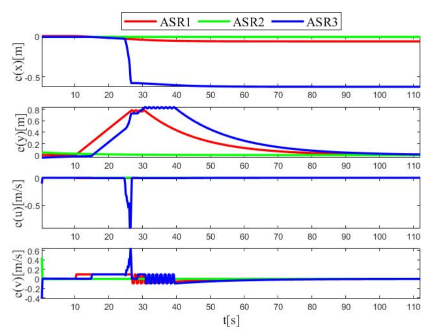


Fig. 9. Formation tracking errors of ASRs in second scenario.

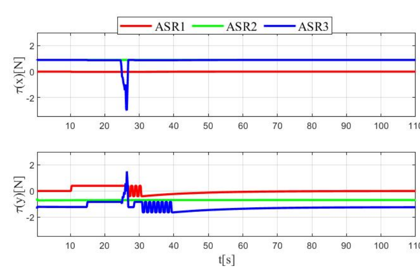


Fig. 10. Control input component of ASRs in second scenario.

V. Conclusion

To solve formation tracking problems of ASRs with moving obstacle avoidance, a collision-free formation trajectory tracking method is designed in this paper. Compared with the traditional formation trajectory tracking controller MPC, the VO-MPC consider the collsion with moving obstacles. Moving obstacles collsion avoidance can be achieved by introducing the velocity constraints generated by VO into the online optimization problem. The simulation results validate the feasibility of the designed algorithm with dynamic obstacle. In addition, uncertainty of dynamics model is planned to be considered in the future reserach.

Acknowledgment

This work was partly supported by National High Tech. Research and Development Program of China (No.2015AA043202), the National Natural Science Foundation of China (61773064, 61503028), the National Key Research and Development Program of China (No. 2017YFB1304401).

References

[1] G. Han, X. Qi, Y. Peng, C. Lin, Y. Zhang, and Q. Lu, "Early warning obstacle avoidance-enabled path planning for multi-auv-based maritime transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2022.

[2] J. W. Choi, A. V. Borkar, A. C. Singer, and G. Chowdhary, "Broadband acoustic communication aided underwater inertial navigation system," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5198–5205, 2022.

[3] R. An, S. Guo, Y. Yu, C. Li, and T. Awa, "Task planning and collaboration of jellyfish-inspired multiple spherical underwater robots," *Journal of Bionic Engineering*, vol. 19, no. 3, pp. 643–656, 2022.

[4] C. Li, S. Guo, and J. Guo, "Performance evaluation of a hybrid thruster for spherical underwater robots," *IEEE Transactions on Instrumentation & Measurement*, in Press, 2022.

[5] X. Hou, S. Guo, L. Shi, H. Xing, and D. Xia, "Improved model predictive-based underwater trajectory tracking control for the biomimetic spherical robot under constraints," *Applied Sciences*, vol. 10, no. 22, p. 8106, 2020.

[6] H. Wei, C. Shen, and Y. Shi, "Distributed lyapunov-based model predictive formation tracking control for autonomous underwater vehicles subject to disturbances," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 8, pp. 5198–5208, Aug. 2021.

[7] J. Erskine, R. Balderas-Hill, I. Fantoni, and A. Chriette, "Model predictive control for dynamic quadrotor bearing formations," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, May 2021, pp. 124–130.

[8] Z. Pan, Z. Sun, H. Deng, and D. Li, "A multilayer graph for multiagent formation and trajectory tracking control based on mpc algorithm," *IEEE Transactions on Cybernetics*, pp. 1–12, 2021.

[9] C.-Z. Liu, L. Li, J.-W. Yong, F. Muhammad, S. Cheng, X.-Y. Wang, and W.-B. Li, "The bionics and its application in energy management strategy of plug-in hybrid electric vehicle formation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 12, pp. 7860–7874, Dec 2021.

[10] D. Panagou, D. M. Stipanovic, and P. G. Voulgaris, "Distributed coordination control for multi-robot networks using lyapunov-like barrier functions," *IEEE Transactions on Automatic Control*, vol. 61, no. 3, pp. 617–632, mar 2016.

[11] P. Wang and B. Ding, "Distributed rhc for tracking and formation of nonholonomic multi-vehicle systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 6, pp. 1439–1453, jun 2014.

[12] H. Yin, S. Guo, and M. Liu, "A virtual linkage-based dual event-triggered formation control strategy for multiple amphibious spherical robots in constrained space with limited communication," *IEEE Sensors Journal*, pp. 1–1, 2022.

[13] R. An, S. Guo, Z. H. H. Liang, and S. Gu, "Uncertain moving obstacles avoiding method in 3d arbitrary path planning for a spherical underwater robot," *Robotics and Autonomous Systems*, vol. 151, p. 104011, may 2022.

[14] M. Sainte Catherine and E. Lucet, "A modified hybrid reciprocal velocity obstacles approach for multi-robot motion planning without communication," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2020, pp. 5708–5714.

[15] R. Han, S. Chen, and Q. Hao, "A distributed range-only collision avoidance approach for low-cost large-scale multi-robot systems," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2020, pp. 8020–8026.

[16] R. Han, S. Chen, S. Wang, Z. Zhang, R. Gao, Q. Hao, and J. Pan, "Reinforcement learned distributed multi-robot navigation with reciprocal velocity obstacle shaped rewards," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 5896–5903, July 2022.

[17] Y. Cho, J. Kim, and J. Kim, "Intent inference-based ship collision avoidance in encounters with rule-violating vessels," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 518–525, Jan 2022.

[18] S. H. Arul and D. Manocha, "Dcad: Decentralized collision avoidance withdynamics constraints for agile quadrotor swarms," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1191–1198, apr 2020.

[19] R. Mao, H. Gao, and L. Guo, "A novel collision-free navigation approach for multiplenonholonomic robots based on orca and linear mpc," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–16, jun 2020.

[20] H. Xing, L. Shi, X. Hou, Y. Liu, and S. Guo, "Design, modeling and control of a miniature bio-inspired amphibious spherical robot," *Mechatronics*, vol. 77, no. 1, p. 102574, 2021.