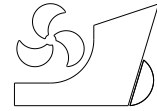


Chuan Liu  
Xianbo Xiang  
Jian Huang  
Shaolong Yang  
Shaoze Zhang  
Xiang Su  
Yunfei Zhang



<http://dx.doi.org/10.21278/brod73105>

ISSN 0007-215X  
eISSN 1845-5859

## DEVELOPMENT OF USV AUTONOMY: ARCHITECTURE, IMPLEMENTATION AND SEA TRIALS

UDC 629.5.072:629.3.05  
Original scientific paper

### Summary

This paper presents the development of autonomy capability for an unmanned surface vehicle (USV). The development mainly focuses on the high-level autonomy on perception, path planning, guidance and control to achieve real sea applications of the USV. First, visual recognition and point cloud data processing techniques are utilized to achieve a real-time perception of the object in the sea environment. Second, detailed path planning strategies are illustrated to plan the easily reachable path for different missions, and the classic guidance and heading controller are adopted to implement the path following algorithm. Subsequently, these autonomy algorithms run in the high-level computer and render the actuator commands for the low-level embedded control system. Finally, sea trials of the USV are conducted by attending the 2020 Zhuhai Wanshan International Intelligent Vessel Competition (IIVC) in Dong Ao Island of South China Sea. The USV accomplish three missions: 1) path following, 2) navigating around the obstacle, and 3) rescuing the drowning. Sea trial results verify the autonomy of the USV in terms of the achieved performances.

*Keywords: Unmanned surface vehicle (USV); High-level autonomy; Perception; Path planning; Guidance and control; Sea trials*

### 1. Introduction

In recent years, unmanned aerial vehicles and ground robots have made great achievements in some searching, mapping, and rescuing missions. However, the development of marine robots, such as unmanned surface vehicle (USV) [1, 2, 3, 4], unmanned underwater vehicle (UUV) [5, 6] and underwater vehicle-manipulator systems (UVMS) [7] is quite challenging due to the complex working environment [8, 9, 10]. Nowadays, there are some competitions of marine robots to promote the autonomy and the applications, by setting some practical missions for students, researchers, and professionals in the ocean engineering

community [11]. Particularly, some field tests for USVs are designed to improve the autonomy and enable the potential real-sea applications [12].

One of the most famous competitions for USVs is the Maritime RobotX Challenge which sets up practical tasks such as underwater search and docking in real sea environments [13]. Since 2014, the contest has been held several times in the USA and Singapore. The World Robotics Sailing Championship also aims to develop potential application technologies such as sweeping and virtual anchoring [14]. Another famous challenge is Microtransat Challenge [15], which requires an ASV to cross the Atlantic from North America to Europe or vice-versa. To promote the development of USVs in China, the first Zhuhai Wanshan International Intelligent Vessel Competition (IIVC) was held in 2020 in Dong Ao Island of South China Sea. By attending the IIVC, team members of Lab of Advanced Robotic Marine Systems (ARMS) in Huazhong University of Science and Technology (HUST), China, developed the autonomy architecture and implemented the algorithms for the dedicated USV used in the competition.



Fig. 1 2020 Zhuhai Wanshan International Intelligent Vessel Competition

The competition puts forward the following missions, the difficulty level of which varies from the easy mission to the difficult one, in order to examine the autonomy of the USV: 1) path following, 2) navigating around the obstacle, and 3) rescuing the drowning. The competition was held in the open sea for four days, regardless of weather and waves.

This USV platform provided by Zhuhai Yunzhou Intelligent Technology Ltd., consists of a catamaran hull, propulsion systems, powering systems, and remote control modules. To achieve navigation and perception capabilities, sensors such as global navigation satellite system/inertial measurement unit (GNSS/IMU), camera, lidar are installed on the USV. An integrated low-level control system is mounted on the USV, which can take over the motion control at any time by the shore-based radio remote control equipment to ensure the USV safety. A high-level computer running autonomy algorithms enables the USV to autonomously perform specific missions through the Ethernet communication with the low-level controller.

In the competition, once the USV is launched from the starting zone, no interactions between human operators and the USV are allowed during the entire running test, except for monitoring the status of the USV remotely via the local wireless network. Hence, autonomy plays an essential role in the autonomous missions. The USV is required to handle all information gathered from sensors to know its status and surrounding environment, plan desired path based on missions, and drive itself along the path autonomously. The high-level autonomy architecture is designed to accomplish different missions, and its performance is tested by integrating the proposed perception, planning, guidance, and control algorithms.

This paper is organized as follows: Section 2 describes the hardware architecture and software structure of the USV system. Autonomous strategies of the USV based on the high-

level autonomy for each mission are described in Section 3, and the sea trial results are illustrated and discussed in Section 4. Finally, Section 5 summarizes the experience and lessons of the competition.

## 2. USV system

The USV system (as shown in Fig. 2) consists of all the hardware and part of the software provided by the competition organizer as well as the high-level autonomy software developed by the participants running on an industrial computer. Another computer runs ground control station (GCS) software to visualize status, tune parameters, and assign missions of the USV. There is a detailed description of the hardware architecture and the software architecture below.



Fig. 2 The USV platform

### 2.1 Hardware architecture

The USV platform with a shore-based monitor system is shown in Fig. 2. A shore LAN is built to connect with the onboard LAN to support Remote debugging. The hardware architecture is described in Fig. 3, which consists of five parts: 1. a catamaran hull, 2. a propulsion power system, 3. a sensor system, 4. a computer system, and 5. a wireless communication and remote controller system.

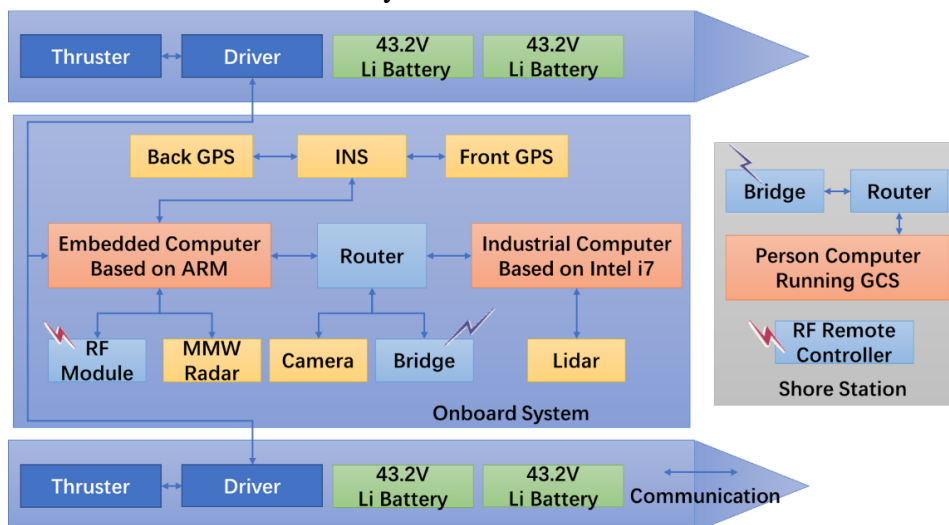


Fig. 3 The hardware architecture

### 2.1.1 Hull

Catamarans have a stronger ability to resist lateral interference than monohulls [16]. Their flexible assembly, greater task capacity, and deck area make them widely used as unmanned engineering boats. The catamaran used consisted of two buoyancy tanks, an upper equipment tank made of carbon fiber, and an aluminum alloy support tube, whose principal dimensions were  $2.5 \times 1.4 \times 1.5$  m. The buoyancy tanks had been installed with a thruster and two batteries. The upper cabin is divided into interconnected front and rear control compartments. A vertical carbon fiber rod and a horizontal one are used to mount the monocular camera and the lidar to improve their perception range.

### 2.1.2 Sensor system

To complete assigned missions autonomously, the status of the USV and surrounding environment information should be extracted from multi-source sensor pieces of information. The GNSS/IMU system is used to estimate the position and orientation in real-time. To perceive the environment, a camera and two radars are installed to identify the color of balls and estimate the positions of balls relative to the USV.

Accurate navigation is the key to accomplishing the mission. Hence, an inertial navigation system (INS) device consists of a 3-axis MEMS gyroscope and a 3-axis MEMS accelerometer. A strapdown inertial navigation algorithm is adopted to provide information on high-frequency pose (i.e., position and attitude) and velocity. However, the status estimation error of strapdown inertial navigation system (SINS) increases rapidly with respect to time, thus the initial state is required to serve as the dead reckoning (DR) origin of calculation [17]. Two GNSS modules are added to the INS device to supply the initial position and orientation. More importantly, an integrated navigation system with GNSS and IMU yields high-precision navigation. The fusion information is sent through the RS232 interface using the standard NMEA-0183 protocol.

A monocular camera with its network server is used to collect the visual environmental information in the front. The camera has a field of view (FOV) angle of  $98.2^\circ$ . The rate of the output video frame is adjusted to 10 fps and the resolution is adjusted to 720p according to the computational requirement.

One of the onboard radars is a 20-channel mechanical lidar which creates 3D imaging by  $360^\circ$  mechanically rotating with 20 oblique laser diodes up and down between  $+8^\circ$  and  $-25^\circ$  inside the housing. Subsequently, the point cloud data at the frequency of 10 Hz is sent through UDP. Although the lidar has a long detection distance and high precision, it has poor performance in extreme weather such as rain, snow, and fog. So, a millimeter-wave (MMW) radar is also installed to deal with this harsh situation.

### 2.1.3 Computer system

There are three computers used in the USV system. The embedded ARM computer is connected with sensors and actuators, which is responsible for forwarding sensor information and processing and distributing commands. It is also connected with a radio frequency (RF) module to receive remote controller commands from the shore station. The other onboard computer is an industrial computer (IPC) based on Intel i7 running the autonomy control software developed by the ARMs team. The computer has two network cards, which are connected to the onboard router and lidar respectively. Besides, a dedicated computer is deployed on the shore station to run remotely monitoring software.

### 2.1.4 Communication and remote control

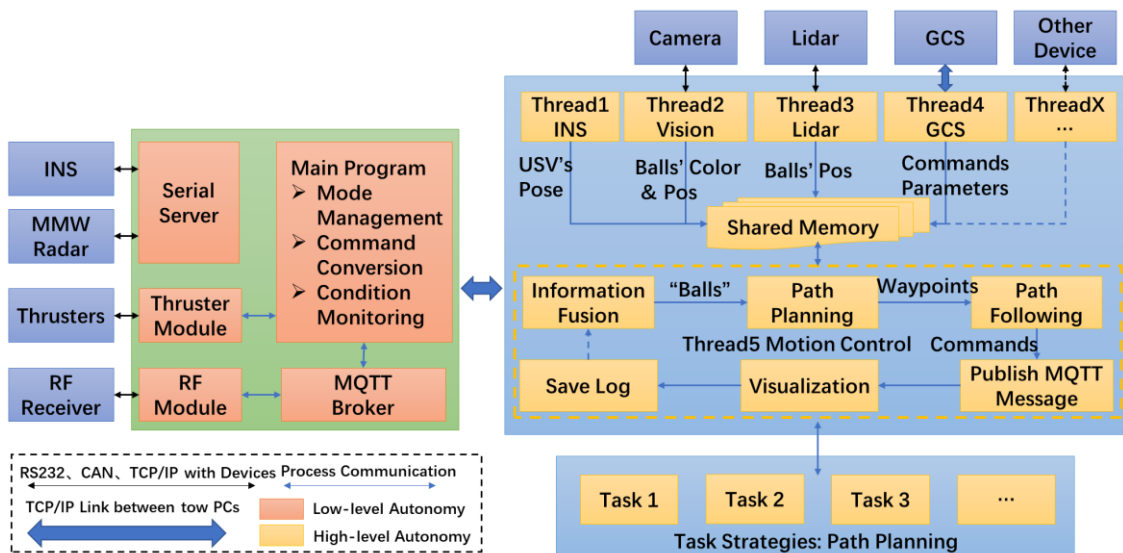
An onboard router connects the embedded computer, the industrial computer, and the webcam into the same local area network (LAN). A pair of wireless bridges (effective distance

about 5km) connects the LAN of each USV in the competition with the shore station LAN, so these three computers mentioned above can communicate with each other. In the early stage of testing, the USV can navigate autonomously through the IPC placed on the shore. While the IPC is installed on the USV, the local software can be debugged and modified through the remote tools.

The other wireless communication link is 2.4 GHz RF used for the remote control. As shown in Fig. 2, there is a switch to change USV working mode, a knob to adjust the max throttle, and two channels to control the motion of the USV. In case of high-level autonomous software occurs with bugs or emergency operations are required, the remote controller could ensure the safe return of the USV.

## 2.2 Software architecture

**Fig. 4** illustrates the software architecture onboard of the USV and the components of the high-level and the low-level onboard software.



**Fig. 4** The software architecture

### 2.2.1 Low-level autonomy software

To ensure the USV always in controlled condition, even if the high-level software in the IPC has faults, a robust software is implemented in the embedded computer developed by the IIVC organizer. The built-in software is not only used to ensure the safety of the USV but also to process the sensor information and send commands to actuators, which provide a good low-level software to enable the competition participants to develop high-level autonomy software.

On the one hand, due to no direct link between the INS/the MMW radar and the IPC, a serial server maps COM ports to TCP ports. The high-level software running on the IPC can obtain the original output data of the two sensors through TCP. On the other hand, an Message Queuing Telemetry Transport (MQTT) broker is used to transfer messages, which distributed messages published by MQTT clients to clients who subscribe to the same topic [18]. Two control interfaces are provided as shown in **Fig. 5**. The message with the topic of /Ctrl has a payload of a float number with range -1 (fully back speed) to 1 (fully forward speed) representing the desired speed, and the other float number with range -1 (hard port) to 1 (hard starboard) representing the desired turn of the USV and a byte number representing the communication priority. The other message with the topic of /Diff has the same frame which uses two float numbers with range -1 (fully forward speed) to 1 (fully reverse speed) to control



the rotation rate of left and right thrusters, respectively. After receiving the two MQTT messages, the main program converts them and then sends commands to the thruster. For users, the former interface is more intuitive and easier to use, but the latter one can make the USV turn in one place which will be used in mission 1 of the competition. The main program also received MQTT messages from the RF module, including the thruster commands mentioned above. Additional mode information is picked up from another message package. If it works in the teleoperation mode, the USV moves according to the commands from the remote controller. Otherwise, the USV moves according to the commands from IPC. As long as the USV is under the range of RF, the operator can take over the control function.

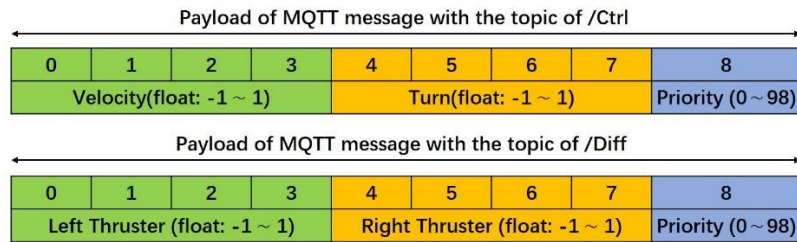


Fig. 5 The MQTT message frame

### 2.2.2 High-level autonomy software

The high-level autonomy software developed by competition participants runs on the onboard IPC. The key to winning the competition is how to develop a simple and feasible software architecture together with a robust autonomous program to drive the USV according to the missions, under the disturbance of light, wind, and wave.

The software is divided into five threads, which use shared memory to transfer information. After initialization, the following five threads are created respectively:

**Thread 1** creates a TCP client, periodically receives INS data, and resolves navigation information according to NMEA protocol.

**Thread 2** uses OpenCV to get real-time images from the webcam through Real Time Streaming Protocol (RTSP) and write the color and the relative position of the detected object into the shared memory.

**Thread 3** processes the point cloud from the lidar to obtain a precise relative position of objects.

**Thread 4** creates a TCP server to wait for the connection from GCS software. Then receive commands from the operator and upload status information to GCS.

**Thread 5** implements the functions of task strategy, path planning and motion control for real application. Firstly, the object information from the webcam and lidar is fused. Secondly, the corresponding path planning strategy is selected according to the mission, so that the desired waypoint sequence can be planned out uniformly. Then, the path following controller calculates actuator commands to drive the USV along the projected path. Next, publish the message to the MQTT broker and save the log.

This architecture is simple and extensible so that a sub-thread can be flexibly added to handle additional equipment, or a new planning strategy can be added for other applications. The sequence diagram of the once common application is shown in Fig. 6. The start command is launched by the remote operator. Then, the motion control thread calculates the control signal of actuators according to scheduled mission and perception information. Meanwhile, the GCS thread uploads USV's status when the communication link is available. Finally, if the mission is accomplished, the remote operator will be informed.

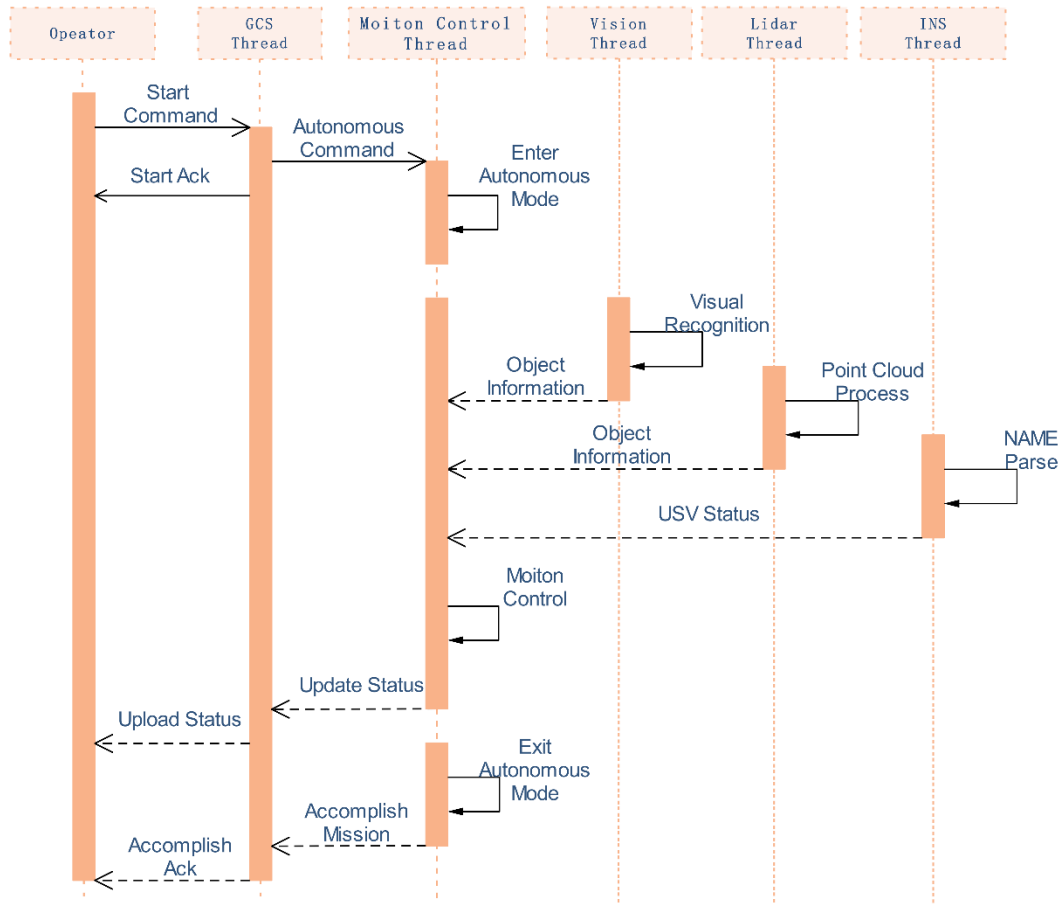


Fig. 6 The sequence diagram of once common application

### 2.2.3 GCS software

The ground control station software shown in Fig. 7 is developed through the Qt software installed on Windows 10. As shown in Fig. 8, the communication protocol between the software running on the IPC and the GCS software is MAVLink V1 [19], which is an open-source protocol widely used in various robots. After customizing the payload, the open-source tool MAVLink Generator is used to generate the API. The software calls the API to resolve messages, displays the USV track on the map and the status via the dashboard and numerical values directly. Furthermore, the software can also pack the task information and system parameters and then send them out. Using GCS software to display USV's motion status and tune parameters made the debugging more efficiently.



Fig. 7 Shore-based monitoring software



Fig. 8 MAVLink V1 frame

### 3. Autonomy control

Based on the hardware and software architectures mentioned in the previous chapter, the high-level autonomy is designed to accomplish the IIVC missions. More specific developments are presented to show the perception, planning, guidance and control in this section.

#### 3.1 IIVC missions

Fig. 9 shows three tasks of the IIVC competition mission, including path following, navigating around the obstacle, and rescuing the drowning.

**Task 1 Path following:** The USV sets off from the departure area and follows a predetermined route precisely. The path consisted of five continuous straight lines determined by five GPS waypoints. Path tracking accuracy is defined as the average deviation from the real-time trajectory to a targeted straight line.

**Task 2 Navigating around the obstacle:** The USV departs from the vicinity of GPS point A and sails to GPS point B. During the voyage, the USV is required to identify and cross the start gate and the end gate represented by black balls. At the same time, when USV recognizes the red ball, it should pass the left, circle around, and to the right of the ball in sequence.

**Task 3 rescuing the drowning:** The USV sails to the disaster area for rescue where yellow balls and black balls are randomly arranged, which are regarded as drowning persons and obstacles respectively. Each circumnavigation of the yellow balls means that a drowning person is successfully rescued.



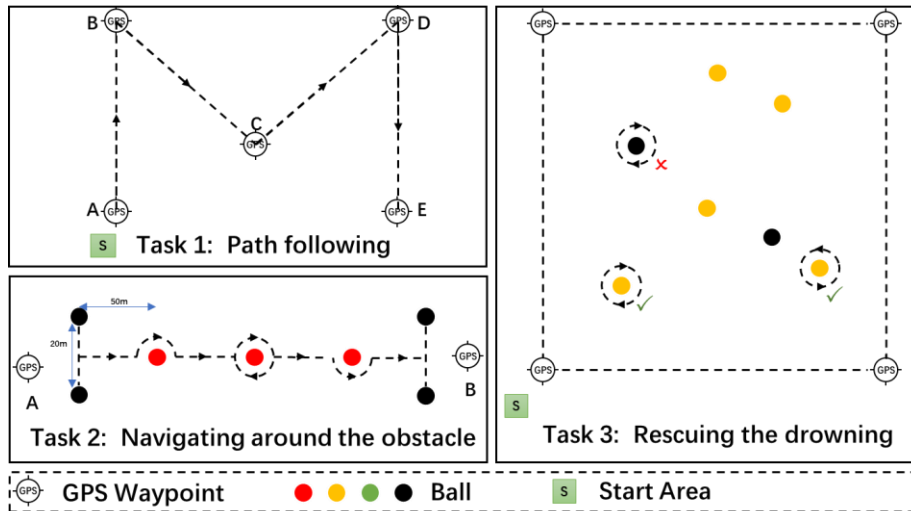


Fig. 9 Three tasks of the IIVC competition

### 3.2 Image processing

Cameras are used to directly detect the presence of balls and the color. To match with the information of the balls detected by the lidar, the rough position of the ball is required.

Fig. 10 shows the flowchart of the visual processing. To get more accurate relative position information, it is necessary to correct the image distortion using the calibration method of Zhang [20]. In the vision thread, the obtained image is first corrected for distortion and then the color space of the image is represented by HSV for facilitating color contrast and extraction [21]. By setting the threshold of each HSV channel, the color of interest is extracted to get the binary image [22]. After a set of morphological transformations, the ball is extracted with little noise. According to the horizontal position of the circle contour in the image and the FOV, the relative orientation of the ball is calculated. A lens imaging principle is used to estimate the relative distance [23]. It is noted that the reflection of the ball must be filtered out because it will result in the mistake of the distance estimation. Although the factors mentioned above are considered, the position estimation accuracy is still low. The ball positioning data from the monocular vision is shown in Fig. 11. The static positioning error is less than 3 meters, but it might be inferred if the dynamic positioning error is large. So, it is necessary to fuse sensing data with the lidar data.

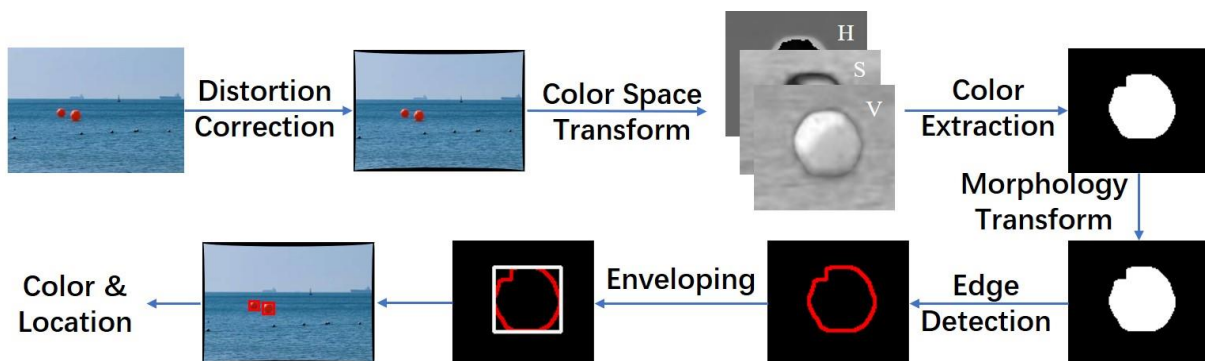
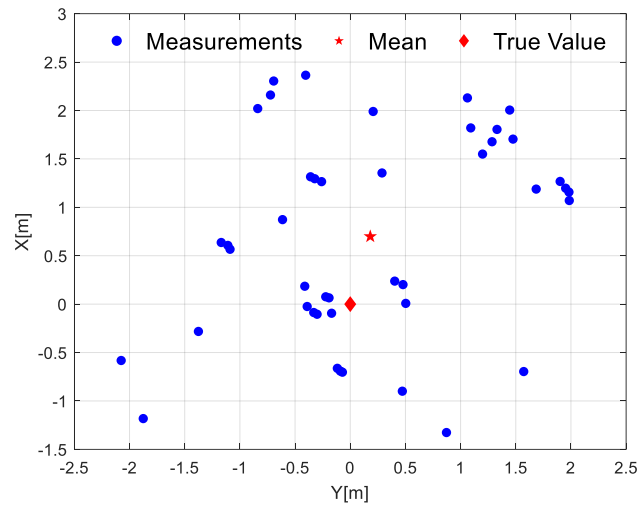


Fig. 10 Flow chart of image processing for target recognition

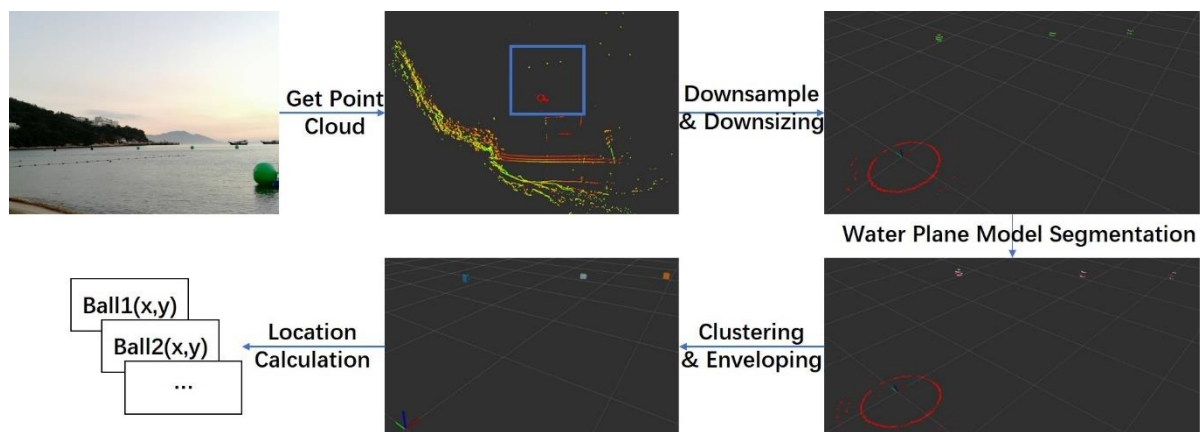


**Fig. 11** Ball positioning from monocular vision

### 3.3 Point cloud processing

Thread 3 receives and processes the point cloud data, and shares the position of balls in shared memory as shown in Fig. 12. The point cloud processing is based on the open-source algorithms from the Point Cloud Library (PCL) [24]. After getting point cloud, a downsampling operation using a voxelized grid approach is performed firstly to reduce the number of points and meanwhile ensure the shape characteristics of the point cloud. The operation can reduce the calculation time of subsequent segmentation and clustering algorithms. By filtering out the far points, the amount of data can be further reduced. Plane model segmentation based on random sample consensus (RANSAC) is used to find the water surface points and filter them out, which improved the clustering effects [25]. After this step, the water surface points are marked as red, and the others are marked as white. The fast clustering algorithm combining Euclidean clustering with the K-dimension tree (KD-Tree) is used to segment the point cloud data [26]. Finally, each group of point clouds is enveloped by cubes and the centers of which are calculated as the centers of balls.

Another important part of perception is the complementary information fusion of balls processed by vision and lidar. Assuming that the position sensed by the lidar is accurate, the information calculated by vision is used for the position matching, in which the threshold of the matched distance is set according to the error of visual positioning. So the color and position of the balls can be obtained at the same time.



**Fig. 12** Flow chart of the point cloud data processing

**Algorithm 1** Path following for multiple lines

---

**Algorithm 1** Path following for multiple lines

---

Input: USV position  $(x, y)$ , Target position  $(x_p, y_p)$ , path tangential angle  $\gamma_p$

- Calculate along-track error  $x_e$  and cross-track error  $y_e$ :

$$x_e, y_e \leftarrow \text{calTrackErr}(x, y, x_p, y_p, \gamma_p)$$

- Calculate desired course angle using LOS guidance:

$$\gamma_d \leftarrow \text{LOS}(y_e, \gamma_p, \Delta)$$

- Implement PID course tracking controller:

$$u_{Turn} \leftarrow \text{PIDControl}(\gamma_d - \varphi, k_p, k_i, k_d)$$

If  $x_e < 5$

- Update desired path:

$$p \leftarrow p + 1; \gamma_p \leftarrow \text{atan2}(y_p - y_{p-1}, x_p - x_{p-1})$$

End if

Output:  $u_{Turn}$

---

### 3.4 Guidance and control

The whole path in the mission can be divided into the straight-line paths and the arc paths, which are also the accessible paths for USVs. The function of the guidance and control part is to calculate the commands to drive the USV moving along the desired path. Especially in the first task, it takes care of the path following accuracy.

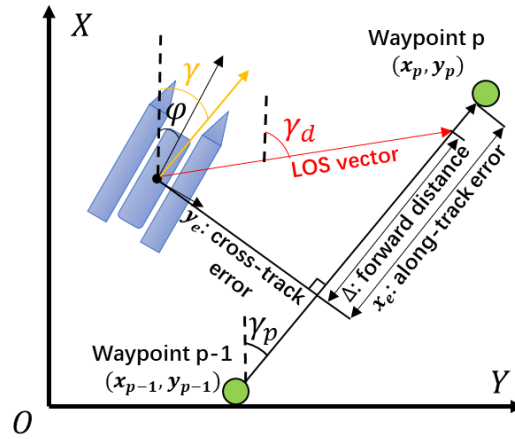
The path following algorithm for multiple lines is shown in **Algorithm 1**, and the key component consists of Line of sight (LOS) guidance algorithm and proportional-integral-derivative (PID) based yaw autopilot [27, 28]. Firstly, the along-track error  $x_e$  and cross-track error  $y_e$  are calculated. Then, to deal with the underactuated properties of the USV, LOS is adopted as the guidance algorithm for line tracking shown in Fig. 13 [29], which converts the path following problem into the course tracking using Eq.(1). In which,  $\gamma_d$  is the desired course,  $\gamma_p$  is the path tangential angle of desired straight path decided by two GPS waypoints,  $\Delta$  is the forward distance which is a parameter to be tuned, usually 3-5 times of the USV length.

$$\gamma_d = \gamma_p + \tan^{-1}\left(\frac{y_e}{\Delta}\right) \quad (1)$$

Then, by using the classic PID yaw autopilot algorithm Eq.(2) and (3),  $u_{Turn}$  renders the actual course  $\gamma$  converges to the desired course  $\gamma_d$ . In this real application, the current heading  $\varphi$  is used to replace the course  $\gamma$ , since the sensor data  $\varphi$  obtained by the INS is relatively stable [30]. In addition,  $u_{Turn}$  is saturated between -1 and 1. In the application, the victory radius is set as 5m. If the along-track error  $x_e$  is less than the victory radius, the USV is regarded as accomplishing following the current path, and the next desired path is updated.

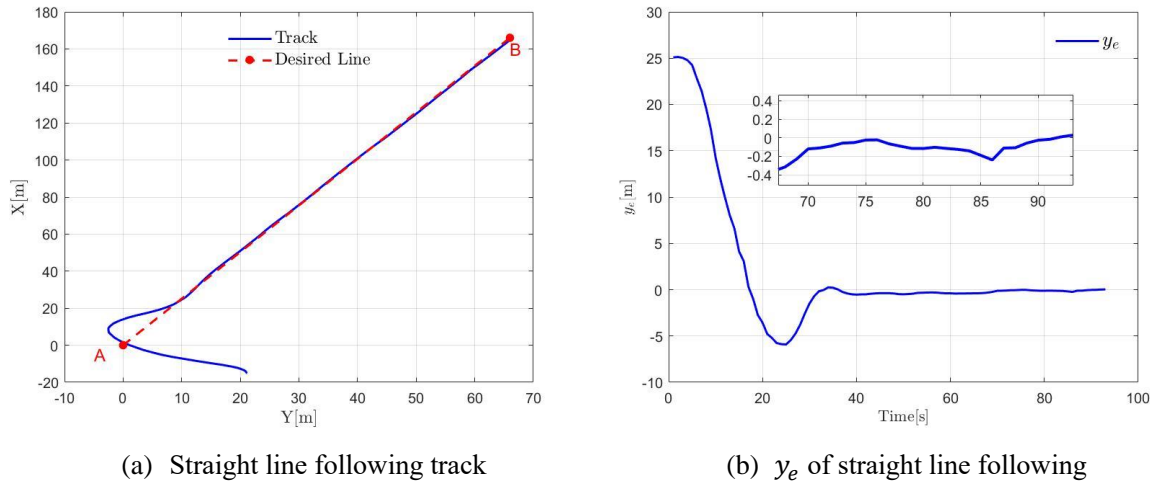
$$u_{Turn} = k_p e_\varphi + k_i \int e_\varphi dt + k_d \frac{de_\varphi}{dt} \quad (2)$$

$$e_\varphi = \gamma_d - \gamma \approx \gamma_d - \varphi \quad (3)$$



**Fig. 13** Line of sight guidance of the USV

Fig. 14 shows the mission results of the straight-line following with a set of fine tuned parameter  $\Delta = 3.8$ ,  $k_p = 0.8$ ,  $k_i = 0.05$ ,  $k_d = 0.2$ . Although the accuracy of straight-line following error is less than 0.2 m, the larger lateral deviation at the corner point will decrease the average accuracy. So, there is room for further optimizations. To achieve easier implementation of a straight-line following algorithm, dedicated strategies for different missions are proposed in form of different waypoint sequences.



**Fig. 14** Straight line following result

### 3.5 Mission strategies

This section describes the planning strategy for each mission to generate the desired waypoint sequence.

**Algorithm 2** The improved path following for multiple lines in IIVC Task 1

---

#### **Algorithm 2** Improved path following for multiple lines in IIVC Task 1

---

Input: USV position  $(x, y)$ , Target position  $(x_p, y_p)$ , path tangential angle  $\gamma_p$  and  $\gamma_{p+1}$

- Calculate along-track error  $x_e$  and cross-track error  $y_e$ :

$$x_e, y_e \leftarrow \text{calTrackErr}(x, y, x_p, y_p, \gamma_p)$$

If  $x_e < 3 m$

- Turn without radius:

$$\text{While}(\text{abs}(\varphi - \gamma_{p+1}) < 10/57.3 \text{ rad})$$

$$\mathbf{u}_{Left}, \mathbf{u}_{Right} \leftarrow \text{turnWithoutRadius}()$$

End while

- Update desired path:

$$p \leftarrow p + 1; \gamma_p \leftarrow \text{atan2}(y_p - y_{p-1}, x_p - x_{p-1})$$

End if

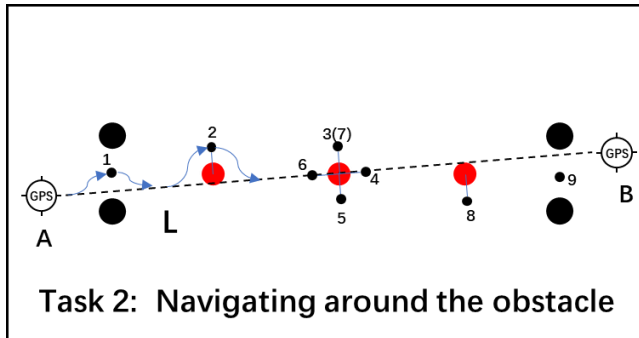
- Continue following path using Algorithm 1

### 3.5.1 Task 1: Path following

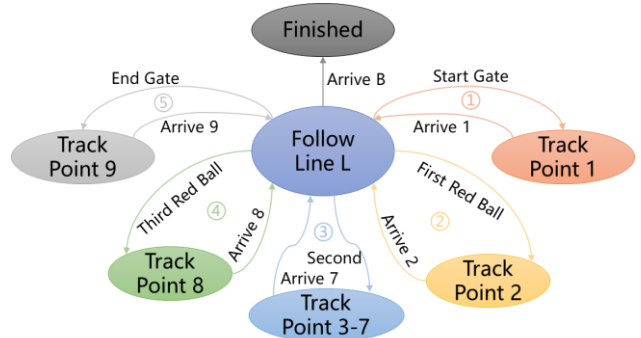
Path following accuracy directly reflects the robustness and performance of USV controllers under the disturbances of wind and wave. In task 1, LOS and PID are directly applied to follow multiple lines. Although the turning path within 5 m around the waypoints is not included in the calculation of the accuracy, there is no guarantee on the turning path in the circle by taking deceleration and early steering actions, especially when the turning angle was acute. Considering the good reverse ability of thrusters, a dedicated strategy for path following in IVC Task 1 is proposed as shown in **Algorithm 2**. When  $x_e < 3m$ , the USV turns without radius by turning a propeller forward and reversing the other propeller through the MQTT message of /diff topic. After turning to the orientation of the next path, the USV switches to the straight-line following using **Algorithm 1** again.

### 3.5.2 Task 2: Navigating around the obstacle

Task 2 requires a precise perception of the start gates, end gates, and yellow balls between them which are achieved by the perceptual processing modules. Then make five corresponding actions according to the rules. As shown in Fig. 15(b), define the major task as following line L determined by the predefined GPS point A and B. If an object is distinguished, the corresponding action is carried out. For example, the USV tracks line L firstly at the beginning of the mission. When the positions of the two black balls are generated via information fusion, waypoint 1 is placed at the center between the two balls. After sailing to waypoint 1, USV tracks line L again. Similarly, waypoints 2-9 will be planned until USV enters the end gate.



(a) The illustration for Navigating around the obstacle



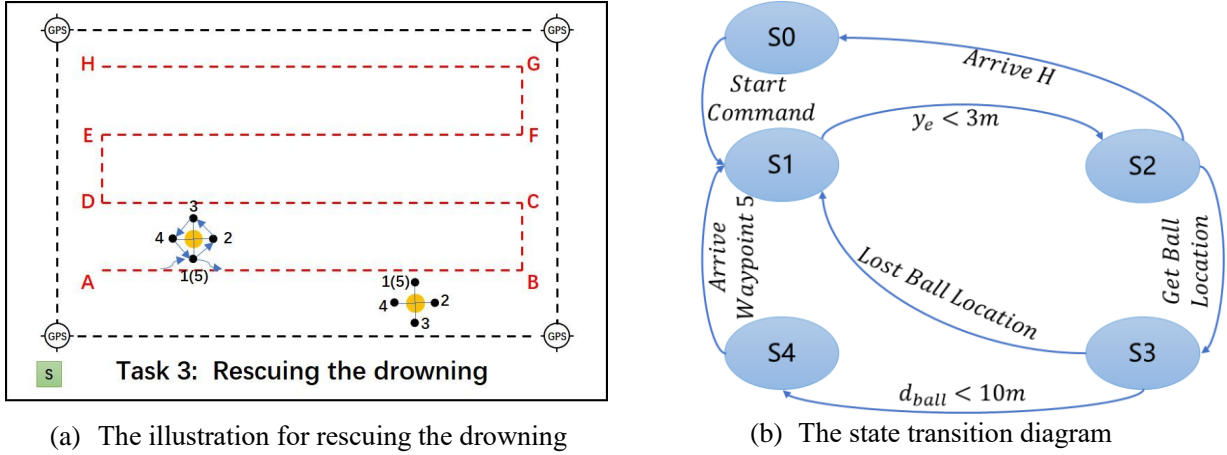
(b) The action transition diagram

**Fig. 15** The strategy for navigating around the obstacle

### 3.5.1 Task 3: Rescuing the drowning

Task 3 required USVs to detect and surround as many as possible yellow balls in a certain time to simulate the searching and rescuing of the drowning. A common lawn-mower path like the red dotted line in Fig. 16(a) is automatically planned in the incident rectangular area, the space of which is determined by FOV and effective distance of camera and lidar. The USV follows the lawn-mower path to search for the drowning. Unlike task 2, the number and position of yellow balls are randomly generated, so all actions of USV cannot be specified in the

preprogrammed way. With the help of the concept of the finite state machine (FSM) [31], the whole task is divided into states as shown in Fig. 16 (b). The definitions of states and trigger regulations are shown in Table 1 and Table 2 respectively. After receiving the start command from the operator, the USV switches into state S2 and runs the program according to the defined action and switching logic. At state S4, the USV needs to judge whether the ball is on the left or right side of the current path, and plan the sequence of waypoints to circle the ball anticlockwise and clockwise, respectively.



**Fig. 16** The strategy for rescuing the drowning

**Table 1** Definition of states and corresponding actions

Index	State	Description
S0	Prepared/Finished	No control action, the actuators in zero position
S1	To Line	Following lawn-mower path
S2	On Line	Following lawn-mower path
S3	To Ball	Perceiving and approaching the yellow ball in real-time
S4	Around Ball	Planning and tracking the waypoints around the ball

**Table 2** Definition of trigger events

Event	Description
Start Command	USV gets the start command
$y_e < 3m$	Cross-track error is less than 3m
Get Ball Location	Perception module locates a yellow ball
Lost Ball Location	In the process of approaching the ball, USV loses its location
$d_{ball} < 10m$	Relative distance of the ball is less than 10m
Arrive Waypoint 5	USV arrives at waypoint 5
Arrive H	USV reaches the last lawn-mower waypoint

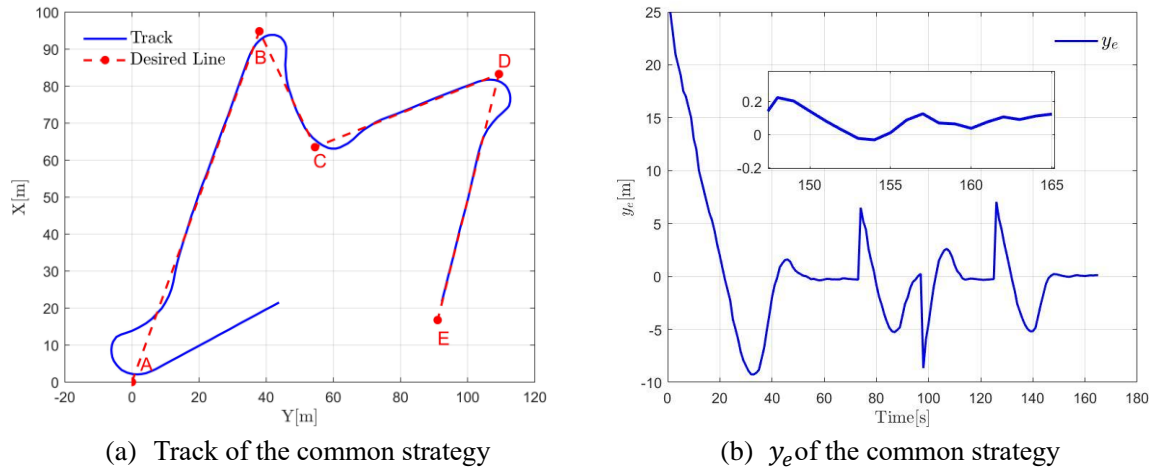
#### 4. Sea trials

In this section, sea trial results from the commissioning stage in the competition are presented, which verify the performance of the USV autonomy in the sea environments.

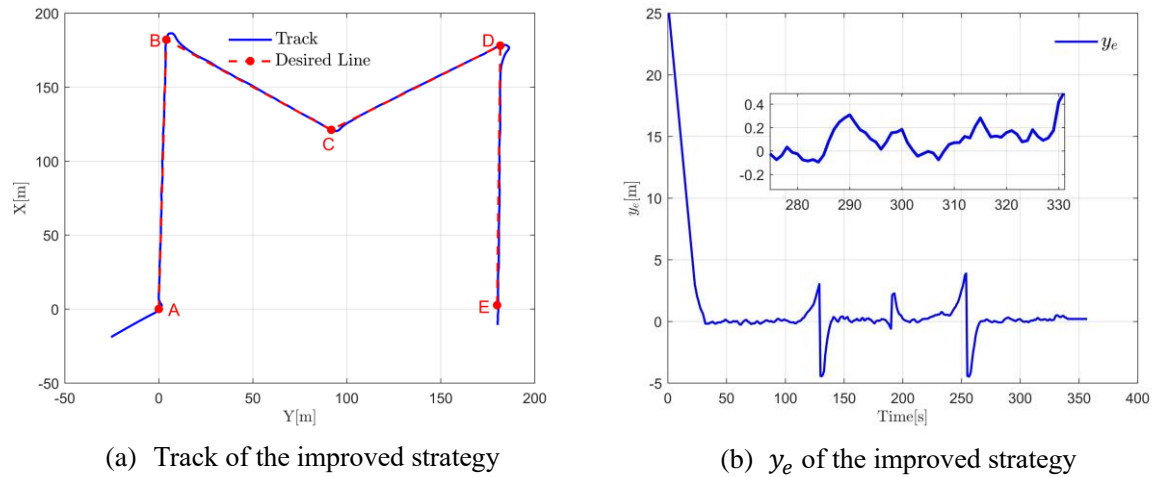


#### 4.1 Task 1: Path following

Task 1 is to evaluate the path tracking accuracy of the USV directly in real sea environments. Both the common strategy and the improved strategy are tested during the commissioning stage. The comparisons of the sea trial results are shown in Fig. 17 and Fig. 18. Both strategies are adopted to track the pre-planned path, in which the tracking error is within 0.2 m in the straight-line following stage. However, using the improved strategy, the offset at the corner waypoint is less than 5 m, which reduces the average tracking error of the whole path following mission to 0.47 m.



**Fig. 17** Trial result of the common strategy for task 1



**Fig. 18** Trial result of the improved strategy for task 1

#### 4.2 Task 2: Navigating around the obstacle

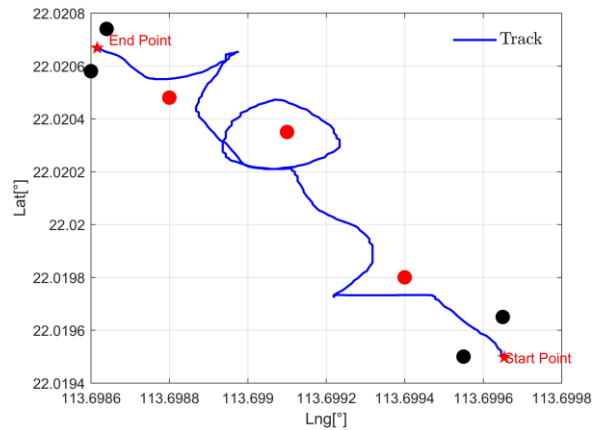
According to the designed strategy for task 2, the USV perceives the corresponding target and takes the corresponding action. Fig. 19(a) is the actual scenario of task 2 and Fig. 19 (b) records the whole trajectory of the USV. As designed, when the USV perceives the first yellow ball, it plans a waypoint to achieve the left avoidance action. After reaching the waypoint, the USV follows the route again. Similarly, the USV orbits around the second yellow ball and executes the right avoidance action.

### 4.3 Task 3: Rescuing the drowning

Task 3 used the FSM to switch the states of sweeping, circling to rescue the drowning and others. Fig. 20(a) shows the sweeping trajectory, in which the USV detects two yellow balls and navigates around them successfully. Because there are no radius and shape requirements, 5 waypoints are planned to generate a circular path around the ball. Fig. 20(b) draws the state transition curve which satisfies the predesigned state switching mechanism. It can be found that there is a moment that state S3 does not switch to state S4. Because the low accuracy of visual positioning causes the failure of information fusion and the loss of ball positioning information. It concludes that the information fusion about balls has a lot of room for improvement.

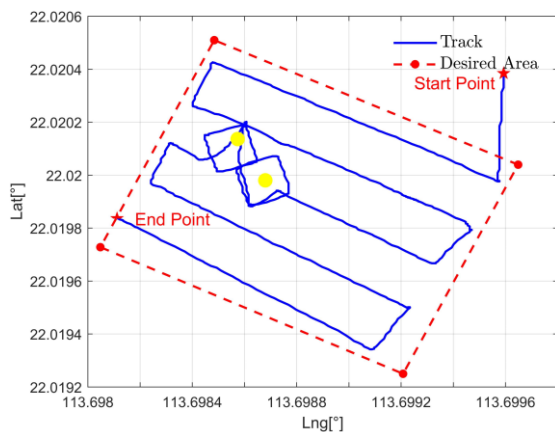


(a) Actual scenario of task 2

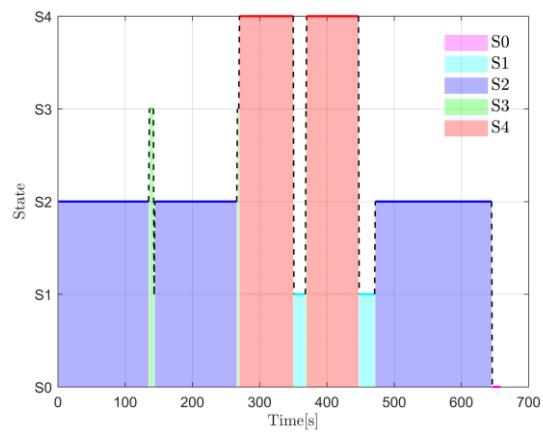


(b) Track of task 2

**Fig. 19** Trial result for task 2



(a) Track of task 3



(b) The transition of state

**Fig. 20** Trial for task 3

## 5. Conclusion

This competition has put forward technical requirements for the high-level autonomy of the USV. A number of experiences of the USV autonomy learned from the competition are shown as follows:

**Open-source tools and libraries:** For the development of an autonomous vehicle, perception, planning, control, communication, and other techniques are indispensable. It's worth mentioning that it is difficult to master the vision or point cloud processing methods proficiently in a short time. However, some useful open-source tools, such as OpenCV, PCL,

and MAVLink protocol provide a quick and convenient solution to implementing these technologies in practice. These open-source library documents and cases are helpful to learn basic operations, explore solutions, and quickly integrate open-source modules into practical applications. However, simple API calls and their combinations could not be well applied, thus proper optimization and improvement are necessary.

**USV architecture:** The architecture including hardware and software onboard, and the shore-based system, is designed in a flexible and reliable way. The separation of the high-level and low-level autonomy layers enables the IIVC participants to design their autonomous control algorithms according to the requirements and quickly access the low-level control system to drive the USV. The stable low-level control mode could be switched into remote control mode at any time to ensure safe navigation in case of emergency. The wireless bridge is composed of several USVs and shore-based control stations under the same LAN. This existing communication architecture enables the development of multi-USV formation, cluster and cooperative operations.

**Perception by vision and lidar:** This competition puts forward the requirement for environmental perceptions. Traditional visual object detection methods are difficult to be used in the outdoor environment, especially in the color extraction step. The light or backlight conditions and the intensity transformation of natural light might affect the threshold settings in the color extraction step. Target identification based on deep learning like Yolo might have a better performance in the outdoor environment. Besides, the visually positioning method in this paper has many side effects which may result in low accuracy. Stereo vision is widely used in visually positioning, which could be used for this competition in the future. Lidar has good detection accuracy in long distances. Facing more accurate application scenarios, the influence of the USV angular motion should be considered.

In summary, this paper presents the development procedures of the USV autonomy architecture by the ARMs Team for the 2020 Zhuhai Wanshan International Intelligent Vessel Contest. The development mainly focuses on the high-level autonomy of the USV to satisfy the requirements of the competition. A number of sea trial results in sea environments show the feasibility and the performance of the autonomy architecture of the USV.

## ACKNOWLEDGEMENT

This work was supported in part by Hubei Provincial Natural Science Foundation for Innovation Groups (under Grant 2021CFA026), in part by National Natural Science Foundation of China (under Grant 52071153), in part by the Fundamental Research Funds for the Central Universities (under Grant 2021yjsCXC012).

## REFERENCES

- [1] Xu, H., Soares, C. G., 2020. Vector Field Guidance Law for Curved Path Following of an Underactuated Autonomous Ship Model, *International Journal of Maritime Engineering*, 162(A3), 249–261. <https://doi.org/10.5750/ijme.v162iA3.1135>
- [2] Li, J., Xiang, X., Yang, S., 2021. Robust adaptive neural network control for dynamic positioning of marine vessels with prescribed performance under model uncertainties and input saturation. *Neurocomputing*, In Press. <https://doi.org/10.1016/j.neucom.2021.03.136>
- [3] Peng, Z., Wang, J., Wang, D., Han, Q., 2021. An overview of recent advances in coordinated control of multiple autonomous surface vehicles. *IEEE Transactions on Industrial Informatics*, 17(2), 732-745. <https://doi.org/10.1109/TII.2020.3004343>
- [4] Volden, Ø., Stahl, A. Fossen, T.I., 2021. Vision-based positioning system for auto-docking of unmanned surface vehicles (USVs). *International Journal of Intelligent Robotics and Applications*. Published online. <https://doi.org/10.1007/s41315-021-00193-0>

- [5] Zhang, J., Xiang, X., Li, W., 2021. Advances in Marine Intelligent Electromagnetic Detection System, Technology and Applications: A Review. *IEEE Sensors Journal*. Published online. <https://doi.org/10.1109/JSEN.2021.3129286>
- [6] Y. Yang, Y. Xiao, T. Li, A Survey of Autonomous Underwater Vehicle Formation: Performance, Formation Control, and Communication Capability. *IEEE Communications Surveys and Tutorials*, 23(2) (2021) 815-841. <https://doi.org/10.1109/COMST.2021.3059998>
- [7] Zhang, Q., Zhang, J., Chemori, A., Xiang, X., 2018. Virtual submerged floating operational system for robotic manipulation. *Complexity*, <https://doi.org/10.1155/2018/9528313>
- [8] Vinayak, P. P., Prabu, C. S. K., Vishwanath, N., Prakash, S. O. Numerical simulation of ship navigation in rough seas based on ecmwf data. *Brodogradnja*, 72(1), 19–58. <https://doi.org/10.21278/brod72102>
- [9] Xiang, G., Xiang, X., 2021. 3d trajectory optimization of the slender body freely falling through water using cuckoo search algorithm. *Ocean Engineering*, 235, 109354. <https://doi.org/10.1016/j.oceaneng.2021.109354>
- [10] Zhang, S., Xiang, X., Yang, S., 2020. Communication network and qos evaluation for formation control of unmanned surface vehicles. *Brodogradnja*, 71(4), 19–37. <https://doi.org/10.21278/brod71402>
- [11] Ferreira, F., Ferri, G. 2020. Marine robotics competitions: a survey. *Current Robotics Reports*, 1, 169-178. <https://doi.org/10.1007/s43154-020-00022-5>
- [12] Wang, Z., Yang, S., Xiang, X., Vasiljević, A., Mišković, N., Nad, Đ., 2021. Cloud-based mission control of usv fleet: Architecture, implementation and experiments. *Control Engineering Practice*, 106, 104657. <https://doi.org/10.1016/j.conengprac.2020.104657>
- [13] Park, J., Kang, M., Kim, T., Kwon, S., Han, J., Wang, J., Yoon, S., Yoo, B., Hong, S., Shim, Y., Park, J., Kim, J., 2017. Development of an unmanned surface vehicle system for the 2014 maritime robotx challenge. *Journal of field robotics*, 34(4), 644–665. <https://doi.org/10.1002/rob.21659>
- [14] Le Bars, F., Jaulin, L., 2015. The world robotic sailing championship, a competition to stimulate the development of autonomous sailboats. *OCEANS 2015-Genova, IEEE*, Genoa, Italy. <https://doi.org/10.1109/OCEANS-Genova.2015.7271767>
- [15] Leloup, R., Le Pivert, F., Thomas, S., Bouvart, G., Douale, N., De Malet, H., Vienney, L., Gallou, Y., Roncin, K., 2011. Breizh spirit, a reliable boat for crossing the atlantic ocean. *Robotic Sailing*, Springer. Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-22836-0\\_4](https://doi.org/10.1007/978-3-642-22836-0_4)
- [16] Castiglione, T., Stern, F., Bova, S., Kandasamy, M., 2011. Numerical investigation of the seakeeping behavior of a catamaran advancing in regular head waves. *Ocean Engineering*, 38(16), 1806–1822. <https://doi.org/10.1016/j.oceaneng.2011.09.003>
- [17] Noureldin, A., Karamat, T. B., Georgy, J., 2012. Fundamentals of inertial navigation, satellite-based positioning and their integration. *Springer Science & Business Media*, Berlin Heidelberg.
- [18] Hunkeler, U., Truong, H. L., Stanford-Clark, A., 2008. Mqtt-sa publish/subscribe protocol for wireless sensor networks. *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWA'08)*, IEEE, Bangalore, India. <https://doi.org/10.1109/COMSWA.2008.4554519>
- [19] Atoev, S., Kwon, K.-R., Lee, S.-H., Moon, K.-S., 2017. Data analysis of the mavlink communication protocol. *2017 International Conference on Information Science and Communications Technologies (ICISCT)*, IEEE, Tashkent, Uzbekistan. <https://doi.org/10.1109/ICISCT.2017.8188563>
- [20] Zhang, Z., 2000. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11), 1330– 1334. <https://doi.org/10.1109/34.888718>
- [21] Schwarz, M. W., Cowan, W. B., Beatty, J. C., 1987. An experimental comparison of rgb, yiq, lab, hsv, and opponent color models. *ACM Transactions on Graphics (TOG)*, 6(2), 123–158. <https://doi.org/10.1145/31336.31338>
- [22] Liu, Z., Chen, W., Zou, Y., Hu, C., 2012. Regions of interest extraction based on hsv color space. *IEEE 10th International Conference on Industrial Informatics*, IEEE, Beijing, China. <https://doi.org/10.1109/INDIN.2012.6301214>
- [23] Bao, D., Wang, P., 2016. Vehicle distance detection based on monocular vision. *2016 International Conference on Progress in Informatics and Computing (PIC)*, IEEE, Shanghai, China. <https://doi.org/10.1109/PIC.2016.7949492>
- [24] Rusu, R. B., Cousins, S., 2011. 3d is here: Point cloud library (pcl). *2011 IEEE international conference on robotics and automation*, IEEE, Shanghai, China. <https://doi.org/10.1109/ICRA.2011.5980567>

- [25] Himmelsbach, M., Hundelshausen, F. V., Wuensche, H.-J., 2010. Fast segmentation of 3d point clouds for ground vehicles. *2010 IEEE Intelligent Vehicles Symposium, IEEE*, La Jolla, CA, USA. <https://doi.org/10.1109/IVS.2010.5548059>
- [26] Klasing, K., Wollherr, D., Buss, M., 2008. A clustering method for efficient segmentation of 3d laser data. *2008 IEEE international conference on robotics and automation, IEEE*, Pasadena, CA, USA. <https://doi.org/10.1109/ROBOT.2008.4543832>
- [27] Saggini, E., Zereik, E., Bibuli, M., Bruzzone, G., Caccia, M., Riccomagno, E., 2014. Performance indices for evaluation and comparison of unmanned marine vehicles' guidance systems. *IFAC Proceedings Volumes*, 47(3), 12182–12187. <https://doi.org/10.3182/20140824-6-ZA-1003.01055>
- [28] Kapetanović, N., Mišković, N., Nađ, D., Bibuli, M., Caccia, M., Bruzzone, G., 2017. Development of model predictive guidance for underactuated marine vehicles: From simulations to experiments. *OCEANS 2017, IEEE*, Aberdeen, UK. <https://doi.org/10.1109/OCEANSE.2017.8084684>
- [29] Fossen, T. I., Breivik, M., Skjetne, R., 2003. Line-of-sight path following of underactuated marine craft. *IFAC proceedings volumes*, 36(21), 211–216. [https://doi.org/10.1016/S1474-6670\(17\)37809-6](https://doi.org/10.1016/S1474-6670(17)37809-6)
- [30] Lu, X., Liu, Z., Chu, Z., 2020. Nonlinear adaptive heading control for an underactuated surface vessel with constrained input and sideslip angle compensation. *Brodogradnja*, 71(3), 71–87. <https://doi.org/10.21278/brod71305>
- [31] Kurt, A., Oezguener, U., 2013. Hierarchical finite state machines for autonomous mobile systems. *Control Engineering Practice*, 21(2), 184–194. <https://doi.org/10.1016/j.conengprac.2012.09.020>

Submitted: 17.12.2021. Chuan Liu, chuanliuno@hust.edu.cn  
School of Naval Architecture and Ocean Engineering,  
Accepted: 07.02.2022. Huazhong University of Science and Technology 430074, Wuhan, China  
Xianbo Xiang\*, Corresponding author: [xbxiang@hust.edu.cn](mailto:xbxiang@hust.edu.cn)  
School of Naval Architecture and Ocean Engineering,  
Huazhong University of Science and Technology 430074, Wuhan, China  
Jian Huang, [jian.huang@yunzhou-tech.com](mailto:jian.huang@yunzhou-tech.com)  
Southern Marine Science and Engineering Guangdong Laboratory (Zhuhai),  
Zhuhai 519080, China  
Zhuhai Yunzhou Intelligence Technology Ltd., Zhuhai 519080, China  
Shaolong Yang, [yangsl@hust.edu.cn](mailto:yangsl@hust.edu.cn)  
School of Naval Architecture and Ocean Engineering,  
Huazhong University of Science and Technology 430074, Wuhan, China  
Zhang Shaoze, [zhangsz123@hust.edu.cn](mailto:zhangsz123@hust.edu.cn)  
School of Naval Architecture and Ocean Engineering,  
Huazhong University of Science and Technology 430074, Wuhan, China  
Xiang Su, [xiangsu@hust.edu.cn](mailto:xiangsu@hust.edu.cn)  
School of Naval Architecture and Ocean Engineering,  
Huazhong University of Science and Technology 430074, Wuhan, China  
Yunfei Zhang, [yunfei.zhang@yunzhou-tech.com](mailto:yunfei.zhang@yunzhou-tech.com)  
Southern Marine Science and Engineering Guangdong Laboratory (Zhuhai),  
Zhuhai 519080, China  
Zhuhai Yunzhou Intelligence Technology Ltd., Zhuhai 519080, China