Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

Obstacle Avoidance for Autonomous Sailboats via Reinforcement Learning with Coarse-to-fine Strategy

Ziyuan Cheng[2], Weimin Qi[2], Qinbo Sun[2], Hengli Liu[2,3], Ning Ding[1,2], Zhenglong Sun[1,2], Tin Lun Lam[1,2] and Huihuan Qian[1,2,†]

***Abstract*— Obstacle avoidance of autonomous sailboats is a complicated task due to big inertia, highly nonlinear motion of sailboat and uncertain disturbance from wind and water current. To deal with the obstacle avoidance problem of autonomous sailboats, we promote a novel method based on reinforcement learning with coarse-to-fine strategy. In this strategy, coarse stage is used to detect the autonomous sailboat roughly in the test environment. Then fine stage is applied to localize the autonomous sailboat accurately. Hereby, the avoidance performance is improved by the transition from coarse to fine. We have verified our algorithms both in simulation and real experiments. With our method, the sailboat avoids the obstacle in higher precision than the method without the coarse-to-fine strategy. The final success rate of obstacle avoidance is near 83% and the rate of reaching goal is 70%. Both simulation results and experiments show that our method is effective.**

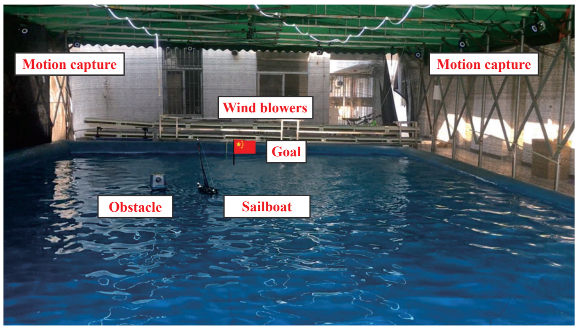


Fig. 1. The overview of autonomous sailboat sails in an obstacle avoidance scenario. The sailboat is sailing windward to reach the goal with the obstacle in the sailboat's heading direction. The motion capture system is providing the state information of sailboat.

I. Introduction

Obstacle avoidance of marine robots has developed a lot during recent years. Researchers realize the collision avoidance of unmanned surface vessels (USVs) with The International Regulations for Preventing Collisions at Sea (COLREGs) protocol [1], [2]. This kind of methods prevent the USVs from collision according to the pre-defined rules. Generally, a complex mathematical model is necessary for the USVs control, which leads to the computation complexity and difficulty of real-time response. Obstacle avoidance procedure will be simplified if robots can learn to realize the avoidance independently. Hence, a concise deep reinforcement learning based method has been proposed for obstacle avoidance of USVs [3]. Simulation results verified the effectiveness of this static obstacle avoidance method. Obstacle avoidance of multi-USVs in motion is still a tough task. Multiple USVs' collision avoidance has been tried with deep Q-learning [4], but the performance could be further improved. Both sparse reward and the high dimension with the state and action bring huge challenges to reinforcement learning (RL) network.

*This paper is partially supported by Project U1613226 and U1813217 supported by NSFC, China, Project 2019-INT009 from the Shenzhen Institute of Artificial Intelligence and Robotics for Society, the Shenzhen Science and Technology Innovation Commission, fundamental research grant KQJSCX20180330165912672 and PF.01.000143 from The Chinese University of Hong Kong, Shenzhen, *R&D* Project of Ministry of housing and urban construction (Grant No.2018-K8-034)

[1] Institute of Artificial Intelligence and Robotics for Society.
[2] The Chinese University of Hong Kong, Shenzhen.
[3] University of Science and Technology of China.
[†] Corresponding author is Huihuan Qian, hhqian@cuhk.edu.cn

Autonomous sailboats consume less energy than other marine robots [5], which play an important role in long-duration ocean applications such as oceanography [6], surveillance [5] and reconnaissance [7]. People have growing interests about autonomous sailboats, the concepts contain competition [8], [9], [10], navigation [2], [11], autonomous sailing based data gathering [12], and et al. Obstacle avoidance is of vital importance in autonomous sailboats' local navigation. The disturbance in the environment is uncertain and the sailboat model is not accurate, which makes it harder for the autonomous sailboat to avoid an obstacle. The RL framework is able to improve the robust and decrease the computation consumption of obstacle avoidance method. Hereby, we delicate design a RL with coarse-to-fine strategy for obstacle avoidance of the autonomous sailboat. The coarse-to-fine perception ameliorates the performance of the RL algorithm for navigation and obstacle avoidance.

In our method, the coarse-to-fine strategy is used to realize accurate localization of an autonomous sailboat to overcome sailboat's big inertia and large delay in motion. In the coarse stage of the reinforcement learning framework, a rough localization of the sailboat and obstacle is obtained. The sailboat can roughly perform navigation and obstacle avoidance task since it is not provided the high precise location information. In this way, the computation complexity is decreased and the task is guaranteed. Once the sailboat is approaching the obstacle, which means the coarse stage can not take vital effect, then it requires the fine stage to take over. The fine stage is working from receiving higher precise location information. The fine stage is also a trained model with more specific obstacle information than the coarse stage. The computation complexity is still not increased but the task

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

precision is increased, which is as what we expected.

The transition between coarse and fine depends on the minimal information resolution. Each stage is a trained model, once the different stage is cascaded connection, the searching space and accuracy will be improved. From the micrography view, it seems the coarse-to-fine operation amplified the location. As the sailboat finished the navigation in fine stage, the sailboat will be back again in the coarse stage with the low accuracy location information. We have verified our obstacle avoidance algorithm in simulation and experiments in a given platform. Both simulation and experiment results show that our method has better obstacle avoidance performance than method without the locally amplified applied.

Our contributions contain three main concepts:
(1) proposed the obstacle avoidance scheme with coarse-to-fine strategy for autonomous sailboats. (2) validate our proposed method in simulation and real environment. The result shows our obstacle avoidance solution is efficient and reliable. (3) the transfer from simulation to real environment has shown that our trained model and scheme can work as we expected.

The rest of paper is structured as follows: Section II formulates the problem. Section III elaborates the reinforcement learning with coarse-to-fine strategy based obstacle avoidance scheme for autonomous sailboats. Section IV presents the experiments and results. Section V concludes the paper.

II. Problem formulation

The main task of the obstacle avoidance procedure is avoiding the obstacle during the autonomous sailboat navigates from a start position p_0 to goal p_g. Then we have the constraint of reaching goal:

$$d_r = ||p_g - p_t||_2 \tag{1}$$

where the p_{tg} is the position of sailboat at time t whiling approaching goal. The p_g is the goal position predefined. d_r is the constraint radius of reaching the goal. In our problem, the d_r is defined as 10 unit pixels in simulation and $0.2m$ in real experiment. Then we have the obstacle constraints:

$$||p_t - p_o|| \geq 0 \tag{2}$$

where p_t is the position of sailboat at time t, p_o is the position of obstacle agent. In the whole navigation process, the sailboat is able to get the position and velocity of an observable obstacle agent based on the perception system. For the sailboat, the position, velocity, heading angle and goal position are accessible. The objective is to minimize the distance from the sailboat position to the goal position.

III. Approach

In this paper, the obstacle is a static agent. The obstacle avoidance task is performed by tuning the adaptive heading angle of autonomous sailboats at each time step. Due to the large inertia and response delay characteristics, a precise location of both obstacle agent and sailboat is of vital importance. With the accurate location information, the sailboat is able to predict the collision previously and response immediately to compensate the big inertia. In our intelligent obstacle avoidance algorithm, the coarse-to-fine framework is used to get more detailed localization. In the coarse stage, we get a rough position of the obstacle agent and sailboat. When it is approaching the obstacle, the transition from coarse to fine is activated. Then, it is locally amplified the map to get fine positions of sailboat and obstacle.

A. The coarse-to-fine strategy based RL

- **State** The state of the RL using in the model represents the coordinate of sailboat. In the model, we use pixels to measure the distance of maze, and set 20 pixels as a unit, which is also the side length of a square grid. We use s_t to represent the current state of a sailboat, and use s_{t+1} to represent the next state of the sailboat. All coordinates are mapped to a square located in the simulation environment.
- **Action** The action in the maze including three directions: *left tack*, *right tack* and *up*, which is corresponding to the real situation of sailboat sailing. Sailboats are not able to run against the wind directly. They can sail windward as a heading angle with respect to the true wind. In our situation, we set the model to choose action up once the step of the other two actions accumulated to four steps, which is satisfied the scenario of our sailboat sails in the testbed. We use a_t to represent the current action of a sailboat, and use a_{t+1} to represent the next action of the sailboat.
- **Transition,** $T(s_t, a_t, s_{t+1})$ is the transition probability from current time step t to the next time step.
- **Reward,** $r(s_t, a_t, s_{t+1})$ The reward represents the score (reward or penalty) obtained by taking the action from the state s_t to the state s_{t+1}, and when the score obtained is positive, it represents the prize, and a negative number is a punishment. Q-learning's brain tends to get higher scores. Thus, each time the sailboat decides to move, it will find the current state in the Q-table and compare the rewards. Then it will obtain the best action for the observable state. The reward distribution is has been shown in Fig.2.
  The sum of immediate reward R_s is calculated with Equation (3),
  $$R_s = \sum \gamma^{t-1} r(s_t, a_t, s_{t+1}) \tag{3}$$
  where $\gamma \in [0, 1]$.
- **The locally amplified RL scheme** To get more accurate localization of the sailboat and obstacle agent, we promote a coarse-to-fine strategy as shown in Fig.3. When the location of the sailboat approaches the 3*3 grid around the obstacle square, the transition operation is activated. It will jump to another 3*3 maze that is equivalent to magnifying two times of the 3*3 grid. Thereby achieving a procedure of coarse-to-fine to achieve reducing the available solution space.

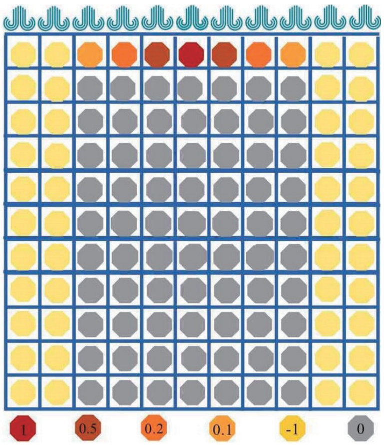


Fig. 2. Training environment and the reward distribution of RL architecture. The wind blows from the top to the bottom. The different color represents the reward value. The goal position is with the highest reward value.

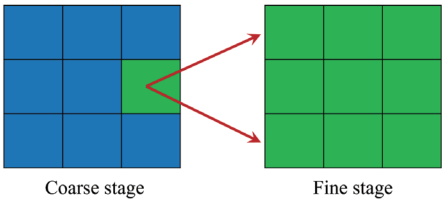


Fig. 3. The coarse-to-fine structure. The blue part is the coarse stage and the green is the fine stage. The arrow represents the transition from coarse stage to fine stage. The resolution of the two stages is different from the whole perspective. Thus, the fine stage can provide more precise location information.

B. Algorithm design of coarse-to fine strategy

Algorithm 1: coarse-to-fine training process. γ is the greedy parameter of Q-learning, which is the possibility of choosing the best choice by searching the Q-table. The reward is decided by current situation s_t, next situation s_{t+1} and current chosen action a_t. First is to initialize s and a, because the sailboat will navigate from bottom to the top, we set the initial state to the left-bottom of maze. At first, there was no reward when the training just starts, so the sailboat will take random action to explore the maze and whenever it get a reward or punishment, it will be recorded into Q-table. After the Q-table has been fulfilled or partly fulfilled, when every episode start, s will be initialized to origin position, then choose action a by searching Q-table to find the highest rewarded action. Then change the current state s to next state s'. When the current state is at the neighborhood position of obstacle, get into the locally amplified maze. The follows steps are similar to the external maze until it gets out of the neighborhood position of obstacle. The loop will stop until sailboat reaches the top which current state s become terminal.

Algorithm 2: obstacle avoidance with coarse-to-fine learning. In the real condition, the $gamma$ has set to be 1, the Q-table has been accomplished. Thus no update is needed. The current state s_o of sailboat will be observed by motion capture, and the new state s_t will also be captured by facilities. The crucial step is to choose action by searching

---

Algorithm 1 Coarse-to-fine learning

Set $\gamma = 0.9$ and reward $r(s_t, a_t, s_{t+1})$;
Initialize $Q(s, a)$ arbitrarily
repeat
  (for episode= $1, 2, ..., N_eps$)
  Choose a from all possible action in this time step
  Take action a, observe r, s'
  Calculate $Q(s, a)$ according to $Q(s, a) \leftarrow r(s, a) + \gamma \cdot \max_{a'}\{Q(s', a')\}$
  $s := s'$
  if sailboat is at the neighborhood position of obstacle **then**
    get into the locally amplified maze
    repeat
      (for episode= $1, 2, ..., n_eps$)
      Choose a from all possible action in this time step
      Take action a, observe r, s'
      Calculate $Q(s, a)$ according to $Q(s, a) \leftarrow r(s, a) + \gamma \cdot \max_{a'}\{Q(s', a')\}$
      $s := s'$
    until sailboat sails out the neighborhood position of obstacle
  else
    not get into the locally amplified maze
  end if
until s is terminal

Algorithm 2 Obstacle avoidance with coarse-to-fine learning

Input: Q table
Output: trajectory $s_{o:t}$
repeat
  Update t, receive new state s_t
  Take action according to Q table and state s_t
  return $s_{o:t}$
until reach the goal position

---

the Q-table.

C. Training process

Initialization: We establish a maze with a size of 11*11 units to realize the training procedure. As shown in Fig.2, the yellow area, distributed on the left and right, is the virtual boundary to create a restricted area for the sailboat to travel. When the sailboat sails into this part of the area, the reward is -1. In this restricted area, the rewards for each location are specially assigned. The rewards on both sides of the target position are gradually decreasing and all locations are with zero rewards except the location of the obstacle. The position of the obstacle is given arbitrarily and its reward is -1. The state is set as the original position of the maze. The first action is given randomly.

The training process has been divided into three different scenarios.

Scenario 1: The model of group-one is mainly used as a control group, the maze used in this group has no obstacle.

We set the reward of two columns in the each edge to -1, and set the reward of the top to 0.1, 0.2, 0.5, 1, 0.5, 0.2, 0.1 from left to right, so as to prevent the sailboat from touching the two sides of the pool, and to find how the path it will be without the obstacle.

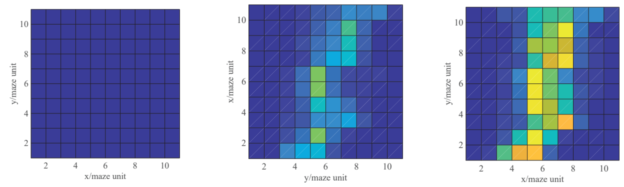
Scenario 2: All settings are the same as the first group, but in order to compare with the model established by the coarse-to-fine method, the second group model set obstacles. At the same time, this model can also be compared with the first group to show the difference of the sailing tendency that is trained through reinforcement learning either there is an obstacle.

Scenario 3: This model uses a coarse-to-fine method. When the square represents the sailboat enters the 3*3 grid which center is the obstacle, the model will jump to another 6*6 grid. That is to magnify two times of the origin 3*3 grid, so the position of the obstacle is more precise. In addition, the model determines the initial position of the sailboat in the 6*6 grid based on the direction in which the sailboat approaches. In this 6*6 maze, the forward rate of the sailboat will be enlarged from the original 4 steps to 2 steps. After the sailboat come out of the 6*6 maze, the model reinitializes the position of the sailboat in the outer maze based on the position of the exit of the inner maze and continues until it reaches the goal position.

Coordinate projection from real word to simulation: Since the number of maze grids used in the model is limited, the coordinates in the trained Q-table are discrete. However, the coordinates of the sailboat in the pool are continuous. In order to make the decision by matching the coordinates with the coordinates in the Q-table, we scale the pool coordinates to find the position of the corresponding coordinates in the maze model, and convergence to the center of the grid, which is the coordinates of the grid in the Q-table. In this way, any continuous coordinates can be represented and then found in the Q-table.

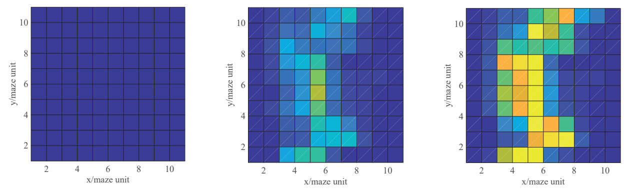
Convergence to the center of the grid: In order to improve the sailing accuracy of the sailboat when avoiding obstacles, we set up a jump of the maze model. When the sailboat reaches one of the grids around obstacles (that is, 1.5m around the center of the real obstacle), we will amplify the 3*3 grid square which the obstacle as the center. Then it can be used to create a 6*6 square. Locate the obstacle position at a new square and set the top-left grid as the coordinate's origin. The location of the sailboat depends on its position in the maze model before it amplified.

On the width of the maze model, we added two columns obstacles on each side and set the reward to -1 to ensure that the real sailboat does not hit the edge of the pool during the sailing windward. When setting the end of the maze model which is reaching the end edge of the pool, we tend to let the sailboat reach the center of the pool edge, so set the reward of the grid at the center of the top to 1 and set the grid on both sides to 0.5, 0.2, and 0.1 by distance.



(a) (b) (c)

Fig. 4. Trained procedure without the obstacle. The probability distribution of sailboat preferred position in the iteration 0, 250 and 500.



(a) (b) (c)

Fig. 5. Trained procedure with the obstacle. The probability distribution of sailboat preferred position in the iteration 0, 250 and 500.

D. Simulation results

We conducted 500 training episodes for each scenarios. The model is trained on the Intel i7-6700HQ@2.6GHz, Nvidia GTX970M and 8G RAM. The training time is approximately 2.5 hours for each scenario.

After training, the sailboat in the first model can successfully reach the target position without touching the edges of the sides, the goal is the center of the top of the maze.

The second model can successfully reach the end of the target position or the adjacent grid of the target position without touching the side edges and avoiding the obstacle.

The third model can avoid the obstacles without touching the edges of the two sides and successfully reach the target position or the adjacent grid of the target position.

Unlike the second model, the third model provides more space for the sailboat. Since the second model tends to move in a space more than 2 grids away from the obstacle, it results in a reduction in its movable space. The sailboat in the third model can be closer to the obstacle, and continue to sail on the grid around the obstacle, which is equivalent to avoid obstacles more subtly. In consequence, the third model can provide a greater navigation space for the sailboat and avoid obstacles more accurately. These two perspectives are as we expected from coarse to fine strategy.

IV. Experiments

A. Experimental Setup

Fig.6 presents the Sailboat Testbed, which contains water pool, wind blowers, light strip, ceiling-installed localization system and local server. The water pool is a sailing area for autonomous sailboat and it is sized with 8m*10m. Wind blowers are able to simulate the natural wind and provide propulsion force for autonomous sailboat. The platform can be controlled remotely, remote-controllable devices contain the electric board, wind blower, light strips and sailboats.

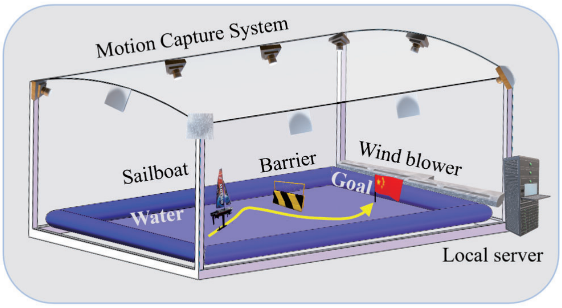
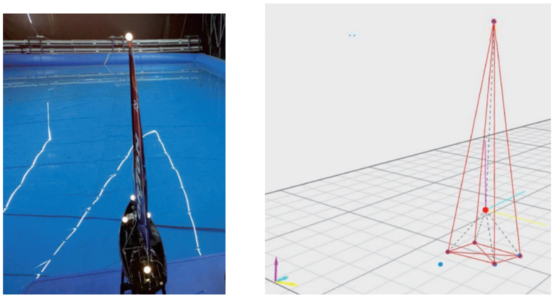


Fig. 6. Architecture of the testbed platform. Equipments of the platform contain the water pool, the motion capture system, the wind blowers, the local server, sailboats, etc. In the figure, the sailboat is trying to avoid the obstacle and towards the goal in the windward case.



(a) (b)

Fig. 7. The motion capture localization system. Fig.7(a) shows the markers' layout of a sailboat and Fig.7(b) is the rigidbody of sailboat produced by the motion capture system.

As shown in Fig.7(a), five markers are installed on the sailboat for localization. Here, a rigidbody needs at least three markers, and more markers on a sailboat will form a more stable rigidbody. A stable rigidbody is necessary for tracking during the sailing of sailboat. Rigidbody of sailboat has been presented in Fig.7(b). The red point in the center is the center of gravity of the rigidbody of sailboat.

As shown in Fig.8, there are four different servers in the local server, which includes motion capture server, database server, serial server and http server. Http server in A can be used to observe and control the sailboats and experiment environment. Motion capture server in B provides the rigidbody location information of sailboats. Server C is the center of the local server system. The MySQL database in C is used to store and fetch the data among different server systems. D is the serial server, which is responsible for the communication between sailboat and MySQL via Bluetooth. A central database has been established by these four servers.

As shown in Fig.9(a), it is a retrofitted autonomous sailboat. Sailboat's length l is 0.465m, and its maximum speed in our testbed is 1.5m/s. Fig.9(a) shows the detailed retrofit structure. Here, the rudder is controlled by a PID controller. The error is the difference value between current heading angle and set heading angle.

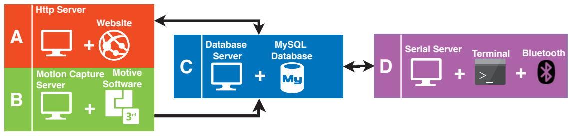
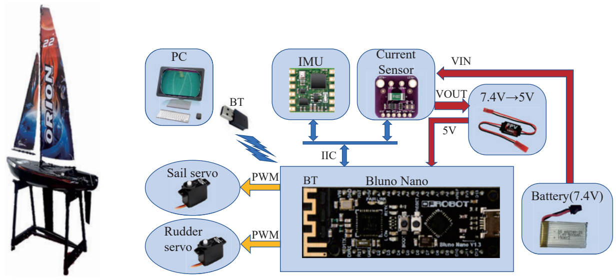


Fig. 8. The relation map of local server system, which includes http server, motion capture server, serial server and database server.



(a) (b)

Fig. 9. Hardware configuration. Fig. 9(b) is the retrofitted sailboat. In the Fig.9(b), the sailboat is equipped with current and voltage sensor module, IMU, Bluno Nano control board DC-DC convert module and battery. The retrofitted sailboat is able to establish communication with remote server system and implement low-level control.

B. Experimental procedure

We have designed three sets of experiments, sailboat sails windward in all experiments. The first set of experiment is about sailboat navigation without obstacle. In this set of experiments, sailboat reaches its goal position after tacking freely several times. There is an obstacle in the second set of experiments, sailboat avoids the obstacle with general Q-learning strategy. The objective of this set of experiments is finding an obstacle-free way from the start to goal. We do not care about how much time it consumes and how far it is when the sailboat starts its obstacle avoidance procedure. In the third set of experiments, coarse-to-fine strategy is applied to localize the obstacle with higher precision. In this set of experiment, we expect less time consumption during the navigation process form star position to goal. We also expect a safety range for sailboat to avoid the obstacle. We performed these three sets of experiments 5-7 times separately, the start position is same at each experiment.

C. Experimental results and discussion

It can be seen that in the first model (scenario 1) in Fig.10, in a group of experiments without obstacles, the sailboat is capable to sail from the bottom to the goal windward. It is valid that our model is stable and effective in sailing navigation.

In model two (scenario 2), the movement of the sailboat is more concerned with avoiding obstacles than the first model as shown in Fig.11, that is, it will be consciously far from the obstacle area. It can be treated as a timid solution, which because the model 2 is only with a coarse stage. The sailboat didn't know the specific location information fo the obstacle. Thus, our experiment result is corresponding to the expected performance.

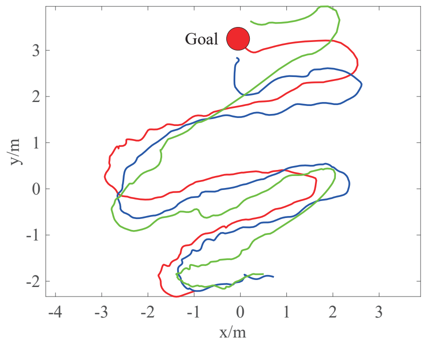


Fig. 10. The experiment results of navigation without obstacle. The sailboat sails after four tackings, and reached the goal successfully. The different color represents the different times of experiments, which validate the model can navigate the sailboat to the goal.

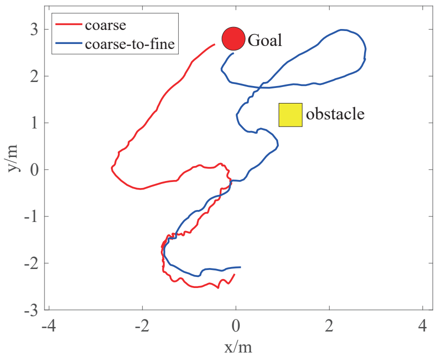


Fig. 11. The experiment results with obstacle to avoid during the navigation process. The yellow square is the barrier and the red circle represents the goal target. The red line shows the experiment with the coarse obstacle avoidance and the blue line represents the experiment with the coarse-to-fine obstacle avoidance. It can be shown that the coarse-to-fine can avoid the obstacle in high precision.

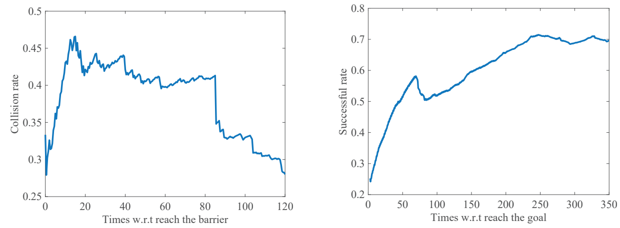
In the experiment of model three (scenario 3), the sailboat avoids the obstacle from the generated path in Fig.11 The distance from the obstacle is closer when the avoidance action is made, which states the sailboats knows more detail obstacle information than in scenario 2. It can be clearly seen that the reinforcement learning has achieved the expected effect on the training of navigating with obstacles, and our coarse-to-fine strategy performs well in improving precise of obstacle avoidance.

The evaluation of our model is shown in Fig. 12(a), The collision rate is decreasing with the times of reach the barrier increased. Based on the statistic result, after 500 iterations, the agent realizes 83% successful avoidance rate (27% collision rate). The increasing part at the beginning results from the sparse obstacle. From the Fig. 12(b), the success rate of reaching the goal is increasing with the times of reaching goal increased. After 500 iterations, the success rate is approaching 83%. It shows that our model performed well at avoiding the obstacle and reaching the target.

V. Conclusions and Outlook

In this paper, we propose a novel obstacle avoidance method based on reinforcement learning with coarse-to-fine strategy. The coarse-to-fine approach contains the coarse stage and fine stage. The performance of obstacle avoidance is roughly if only the coarse part is used to decide the rough location of the sailboat. Thus, the fine part is applied to achieve more accurate localization and better avoidance performance. Both simulation and experiment results have verified the availability and effectiveness of our obstacle avoidance scheme.

In the future, we will adopt deep neural networks to realize more smooth obstacle avoidance of autonomous sailboat. The deep neural network is helpful for optimal policy generation in operating the high-level task for robots.



(a) (b)

Fig. 12. The performance of collision rate and the success rate with reaching the goal. The result is shown in the 500 iterations. The collision rate is decreasing with more interactions, which represents the model has learned to avoid the obstacle. The successful rate is increasing with more interactions, which depicts the model have learned to navigate to the goal.

References

[1] X. Sun, G. Wang, Y. Fan, D. Mu, and B. Qiu, “Collision avoidance of podded propulsion unmanned surface vehicle with colregs compliance and its modeling and identification,” *IEEE Access*, vol. 6, pp. 55 473–55 491, 2018.

[2] Y. Kuwata, M. T. Wolf, D. Zarzhitsky, and T. L. Huntsberger, “Safe maritime autonomous navigation with colregs, using velocity obstacles,” *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 110–119, 2014.

[3] Y. Cheng and W. Zhang, “Concise deep reinforcement learning obstacle avoidance for underactuated unmanned marine vessels,” *Neurocomputing*, vol. 272, pp. 63–73, 2018.

[4] H. Shen, H. Hashimoto, A. Matsuda, Y. Taniguchi, D. Terada, and C. Guo, “Automatic collision avoidance of multiple ships based on deep q-learning,” *Applied Ocean Research*, vol. 86, pp. 268–288, 2019.

[5] N. A. Cruz and J. C. Alves, “Autonomous sailboats: An emerging technology for ocean sampling and surveillance,” in *OCEANS 2008*. IEEE, 2008, pp. 1–6.

[6] C. Sauzé and M. Neal, “Design considerations for sailing robots performing long term autonomous oceanography,” in *International Robotic Sailing Conference*. Centrobot, 2008, pp. 21–29.

[7] R. Stelzer and K. Jafarmadar, “Communication architecture for autonomous sailboats,” in *Proceedings of International Robotic Sailing Conference*, 2009, pp. 31–36.

[8] F. Rejón-Guardia, M. A. García-Sastre, and M. Alemany-Hormaeche, “Sailboat race events: Exploring the effects of a regatta in the balearic islands,” in *Trends in Tourist Behavior*. Springer, 2019, pp. 35–51.

[9] I. Bouty and C. Drucker-Godard, “Managerial work and coordination: A practice-based approach onboard a racing sailboat,” *Human Relations*, vol. 72, no. 3, pp. 565–587, 2019.

[10] “The microtransat,” http://www.microtransat.org/.

[11] R. Stelzer and T. Pröll, “Autonomous sailboat navigation for short course racing,” *Robotics and autonomous systems*, vol. 56, no. 7, pp. 604–614, 2008.

[12] N. Ammann, F. Hartmann, P. Jauer, J. Krüger, T. Meyer, R. Bruder, and A. Schlaefer, “Global data storage for collision avoidance in robotic sailboat racing–the world server approach,” in *Robotic Sailing*. Springer, 2011, pp. 157–166.