Proceedings of the 2007 IEEE Symposium on Artificial Life (CI-ALife 2007)

Cooperation of Multiple Fish-like Microrobots Based on Reinforcement Learning

Jinyan Shao, and Long Wang

Abstract— **This paper is concerned with cooperative control of a kind of multiple fish-like microrobots. Most of previous work on multi-robot cooperation is focused on the terrestrial robots and seldom deals with underwater applications. In fact, the tasks in hydro-environment is more challenging than those in ground circumstances and need the cooperation of robots much more. In this paper, we investigate this problem in the framework of an adversarial game with several underwater microrobots. A fuzzy reinforcement learning approach is adopted to acquire cooperative behavior and a behavioral hierarchical architecture is proposed. We conduct extensive experiments to verify the effectiveness of the proposed algorithms.**

I. INTRODUCTION

Cooperation of multiple robots has been studied more and more in recent years. The most important reason to use multiple robots is to solve some complex and difficult problems in real word, such as engineering design, intelligent search, military missions, and so on. In multi-robot systems, every robot attempt, through their interaction, to jointly solve tasks or to maximize utility. If the environment is uncertain and dynamic, the interaction will be quite complicated. In this circumstance, hand-coding based cooperation might be not feasible. Instead, robot learning might be another good way.

Robot learning is based on the idea that a robot can be trained using incomplete data and then allowed to rely on its ability to generalize the acquired knowledge to novel environments. In all the robot learning algorithms, reinforcement learning(RL) is one of the most attractive learning framework with a wide range of application areas [1]. RL does not require training data and estimates how good or how bad it is for the robot to take a action in a state through giving reinforcement signals(rewards and punishment). RL has been applied to various behavior-based systems and achieved successful results [2] [3].

In this paper, we investigate the cooperative control of a kind of underwater microrobots based on reinforcement learning. Most of the previous work on multi-robot learning is concerned with ground mobile robots, and seldom deals with underwater robots. As we know, compared with ground environment, the underwater circumstance is more complex. It is hard for the designers or researchers to operate the robot by hand coding or based on some experiences. In such a situation, a preferable solution is to let the robots learn by themselves to take the optimal actions in given states. The robot employed in this paper is a kind of biomimetic fish-like microrobot and the task is a cooperative water-polo in an adversarial environment. Combined with behavior-based approach, fuzzy reinforcement learning is utilized to enable the microrobots cooperate and coordinate with each other to achieve the common objective.

The paper is briefly outlined as follows. In Section II, the fish-like micro-robot model is described and the cooperative task is introduced. Section III presents the behavior-based hierarchical architecture for cooperation and a survey of reinforcement learning. In Section IV, the fuzzy reinforcement learning algorithm is adopted to achieve a self-learning cooperation strategy in an adversarial environment. Section V gives the experiment results and discussions. Finally in Section VI, we conclude the paper.

II. FISH-LIKE MICROROBOT MODEL AND TASK DESCRIPTION

A. Description of the fish-like microrobot

Fig. 1 shows the fish-like microrobot prototype and its mechanical configuration. The robot's head and fore-body are molded using fiberglass, in which installed the electrical components: the control module, additional peripherals and power supply. Its rear-body consists of three linked DC servomotors which are wrapped by waterproofed skin. A lunate foil is attached to the last link, which serves as the tail fin. The microrobot mimics the carangiform swimming mode of natural fish and the thrust is mainly generated by undulatory motion of the rear flexible body and the tail.

For carangiform swimming mode, Lighthill [6] suggested a travelling-wave model:

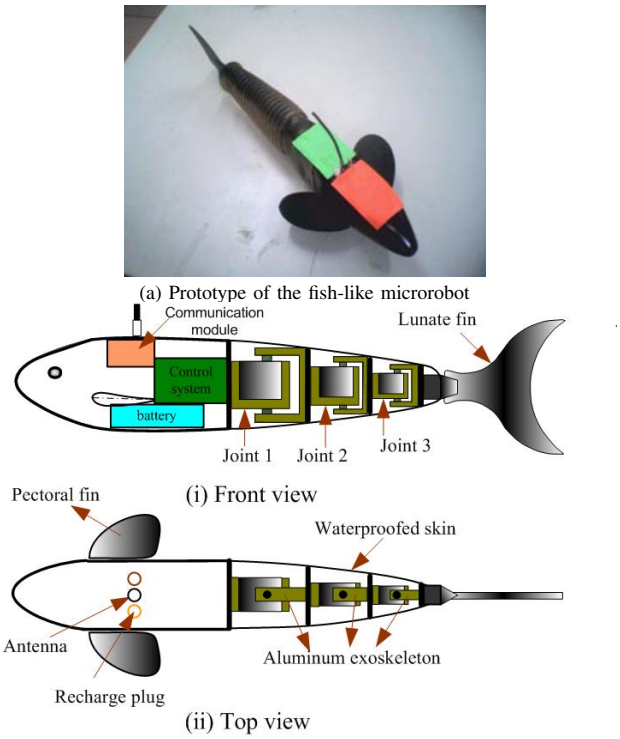
$$y_{body}(x,t) = (c_1 x + c_2 x^2)\sin(kx + \omega t) \qquad (1)$$

where y_{body} denotes the transverse displacement of the fish body, x represents the displacement along main axis, k indicates the body wave number $(k = 2\pi/\lambda)$, λ is the body wave length, c_1 is the linear wave amplitude envelope, c_2 is the quadratic wave amplitude envelope and ω is the body wave frequency (ω=$2\pi f$=2π/T).

For simplicity and feasibility in engineering design, we consider the discrete model of (1). We separate time variable t from y_{body}. Then the travelling body-wave is decomposed into two parts: the first part is the time-independent spline curve sequences $y_{body}(x,i)$ $(i = 0,1,\cdots,M-1)$ in an oscillation period, which is described by (2), and the second part is the time-dependent oscillating frequency f, which

This work was supported by NSFC (10372002 and 60528007) and National 973 Program (2002CB312200)

The authors are with Center for Systems and Control, Department of Mechanics and Engineering, Peking University, Beijing, 100871, P. R. China email: jyshao@mech.pku.edu.cn



(a) Prototype of the fish-like microrobot

(b) Mechanical configuration of robot fish.

Fig. 1. Fish-like microrobot prototype and its mechanical configuration .

is described as the times of recurring oscillations in a unit time interval.

$$y_{body}(x,i) = (c_1 x + c_2 x^2)\sin(kx \pm \frac{2\pi}{M} i) \qquad (2)$$

where i denotes the ith variable of the sequences $y_{body}(x,i)$, M, called body wave resolution, stands for discrete degree of the travelling wave, the signs "$+$" and "$-$" represent different initial moving directions, which are determined based on different initial values.

When the robotic fish swims, its speed is adjusted by modulating the joints' oscillating frequency f, and its orientation is tuned by different joints deflectionθ. Regulate f and θ can produce a series of swimming gaits for the microrobot.

Table I presents basic technical parameters of the fish-like microrobot.

B. Task Description

Though a microrobot bears the virtue of agility and maneuverability, it is often incapable when accomplishing some complex missions. In this circumstance, cooperation of multiple microrobot might be required as a solution. Cooperative control has been demonstrated in many ground robot systems. In this paper, we deal with the underwater case and design a cooperative task *Water Polo 2vs1* as shown in Fig. 2. In a quadrate tank, there are three fish-like microrobots and a polo. The two robots in the left side are

TABLE I

TECHNICAL PARAMETERS

Quantity	Characteristic
Size (L ×W× H)	∼400mm×40mm×78mm
Weight	∼0.5 Kg
Number of the links	3
Maximum oscillating frequency	2 Hz (in calm water)
Length of oscillating part	178 mm
Maximum advancing speed	∼ 0.42 m/s
Minimum turning radius	∼ 200 mm
Maximum input torque	3.2 kg × cm
Working voltage	4.8 V
Actuator mode	DC servomotor
Control mode	Wireless (433 MHz)

teammates and called Offensive Side. The robot in the right side is Defensive Side. If the Offensive Side push the polo into the gate G in the given time, they succeed. Otherwise, the Defensive Side wins.

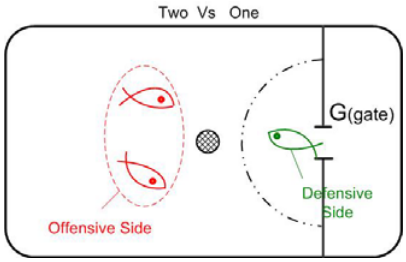


Fig. 2. Two offensive robots Vs one defensive robot.

This task is inspired by the RoboCup which is now a canonical platform for multi-agent cooperation. Compared with the ground circumstance, the hydro-environment is more complicated and uncertain, which makes the cooperative task more difficult to achieve. In this task we use two robots for the Offensive Side and one for the Defensive Side. That is because through experiments, we find, in the water environment, attacking is more difficult than defending.

III. CONTROL ARCHITECTURE FOR COOPERATION

For the two sides in the task, we adopt different control mechanism: the Defensive Side takes a fixed reactive defending strategy which will be presented in Section V; while the Offensive Side will learn to cooperate with each other and achieve the attacking commonly. In this section, we fucus on the organizing approach of the Offensive Side. A behavior-based hierarchical architecture is used as a basic framework as shown in Fig. 3. Four levels are planned. The first level is role assignments. Since in the attacking task, two subtasks should be accomplished: pushing the polo and blocking the defense robot. Based on the subtasks, we defined two roles, namely Attacker and Blocker. Each role is composed of several behaviors. There are totally two kinds of behaviors: reactive behavior or deliberative behavior.

Reactive behaviors display response in direct to environment stimulation, while deliberative behaviors involve some sort of cooperative intention. Here, a behavior represents action or action sequence with intention. The behavior is a reaction to some stimulus from the environment, or a way of acting. For the fish-like microrobot, an action means a gait sequence in a certain mode, which can be obtained from the perspective of AI or bionics through extensive experiments.

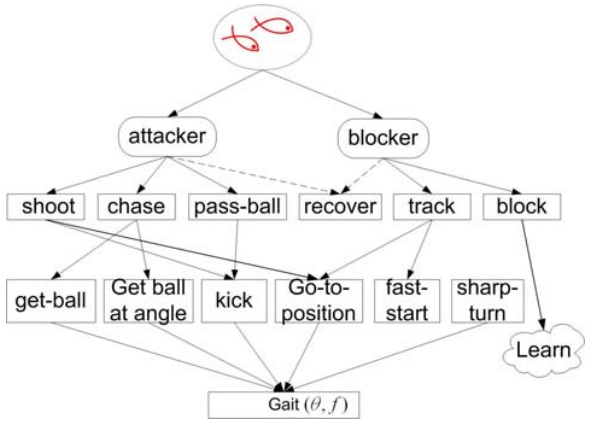


Fig. 3. Behavioral hierarchical framework.

In principle, robot learning can be applied into every level of the architecture. Considering the property of this task and the efficiency, we adopt learning mechanism only in some deliberative behaviors in the behavioral level. For every behavior, we can design a learning controller. But in order to improve the efficiency, we will define Reactive Behaviors by hand-coding on a priori knowledge and use learning method only in the deliberative behavior. Here we choose the block behavior, which aims to prevent the defense from touching the polo and help the attacker hold polo. This behavior is not easy to defined by hand-coding method, since the hydrodynamics often makes the environment not like what people imagine. So we prefer the robot learns by itself to become a good blocker.

A. A survey of reinforcement learning

Fig. 4 shows the standard reinforcement learning model of agent-environment interaction. It consists of

1) discrete set S of environment states s
2) discrete set A of agent actions a
3) scalar reinforcement learning signal r

On each step of interaction the agent percepts the current state s of the environment, then chooses an action a. The action changes the state of the environment to a new state s', and the value of this state transition is evaluated through a scalar reinforcement signal r. The agent should choose actions that tend to maximize the long-run sum of values of the reinforcement signal.

In RL the following functions are defined:

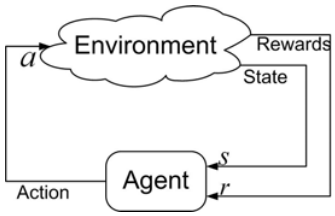


Fig. 4. The standard reinforcement learning model

- $R(s,a)$, the expected value of rewards; a function from state-action pairs to numeric values, i.e., $R : S \times A \rightarrow r$
- $T(s,a,s')$, the state transition function from state-action pair to state, $T : S \times A \rightarrow \pi(S)$. It stands for the probability of making a transition from state s to state s' using action a

In order to evaluate how good it is to take a action a in a state s, a value function $V^*(s)$ is defined. The optimal value of $V^*(s)$ can be given by

$$V^*(s) = \max_a [R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') \cdot V^*(s')] \quad (3)$$

where $\gamma \in [0,1]$ is discount factor, $\gamma \sum_{s' \in S} T(s,a,s') \cdot V^*(s')$ represents the infinite discount sum of rewards which is obtained assuming that the agent always selects the optimal action in state s'. Given the optimal value function $V^*(s)$ the optimal policy $\pi^*(s)$ can be obtained as

$$\pi^*(s) = arg\max_a [R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') \cdot V^*(s')] \quad (4)$$

In most circumstances, the optimal policy is unknown. In that case, the value function can be learned using Temporal-difference(TD) learning. The simplest TD update method, known as TD(0) can be presented as

$$V(s) = V(s) + \alpha[r + \gamma V(s') - V(s)] \quad (5)$$

- α, learning rate, $\alpha \in$ [0,1]
- r, instantaneous reward
- γ, discount factor
- $V(s)$, estimated value function of the current state s
- $V(s')$, estimated value function of the current state s'

This algorithm looks only one step ahead when adjusting value estimates.

IV. COOPERATIVE BEHAVIOR ACQUISITION BASED ON FUZZY REINFORCEMENT LEARNING

In this section, reinforcement learning is adopted to acquire the block behavior for the Offensive Side.

A. Fuzzification of States and Actions

In reinforcement learning, discrete state set and action set are required, while in our task, the state space and action space are continuous. It is unlikely to store all states and actions in a limited memory. Therefore, generalizing the continuous states and actions with respect to a particular method seems to be a more appropriate solution. Here we

adopt fuzzy logic to discretize the state and action space. Robots' attitudes and the polo's position were used as state variables. These variables were fuzzificated into several levels for the fuzzy inference system. All input variables are two-dimensional.

First, as shown in Fig. 5, we make coordinates transformation using polo as the origin and the direction from the polo to the gate(G) as the new x coordinates. In this new coordinates, we can get the membership function representing the position states of the Blocker and the Defense as shown in Fig. 6. Combined the positions of x coordinates and y coordinates, we get 9 position states for both the Blocker and Defense. For every position state, we obtain 8 orientation states using fuzzy logic as illustrated in Fig. 7.

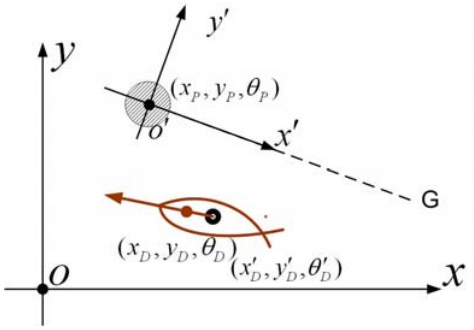


Fig. 5. Coordinates transformation.

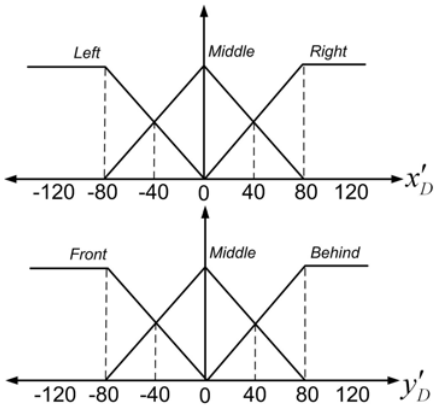


Fig. 6. Membership function representing the robot's position.

The presentation of the position state of the robot are:

- *LEFT-FRONT*
- *LEFT-MIDDLE*
- *LEFT-BEHIND*
- *MIDDLE-FRONT*
- *MIDDLE-MIDDLE*
- *MIDDLE-BEHIND*
- *RIGHT-FRONT*
- *RIGHT-NIDDLE*

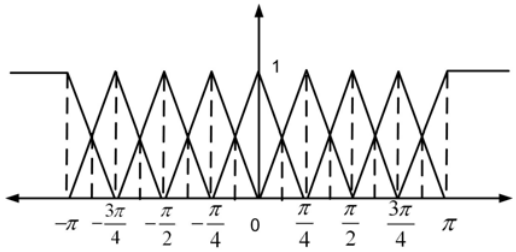


Fig. 7. Membership function representing the robot's orientation.

- *RIGHT-BEHIND*

For the orientation state, there are

- *FORWARD*
- *FORWARD-RIGNT*
- *RIGHT*
- *BACKWARD-RIGHT*
- *BACKWARD*
- *BACKWARD-LEFT*
- *LEFT*
- *LEFT-FORWARD*

So for both the Blocker and Defense, there are $9 \times 8 = 72$ states, and for the blocking behavior, there are totally $72 \times 72 = 5184$ states as shown in Fig. 8.

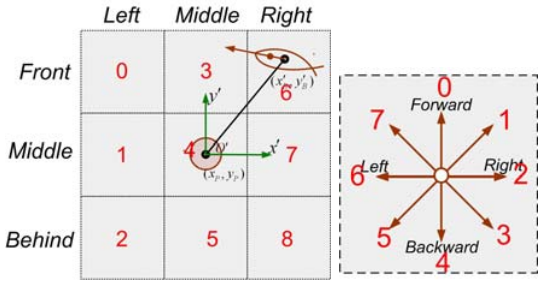


Fig. 8. Representation of state.

For the action space, according to the dynamics of the microrobot, we choose its orientation as the fuzzy variable. After being fuzzificated as in Fig. 9, five actions are obtained: *SHARP-LEFT*, *LEFT*, *FORWORD*, *RIGHT*, *SHARP-RIGHT*.

B. Q-learning process

We use an off-policy TD(0) algorithm Q-learning. The standard one-step Q-learning is defined by

$$Q(s,a) = Q(s,a) + \alpha[r + \gamma \max_a Q(s',a) - Q(s,a)] \quad (6)$$

where the parameters are similar to those in (5). In the learning process:

1) Let α be large at initial state of learning and change to be smaller as the learning progress runs.
2) Let γ be small at initial state and make it increase gradually.

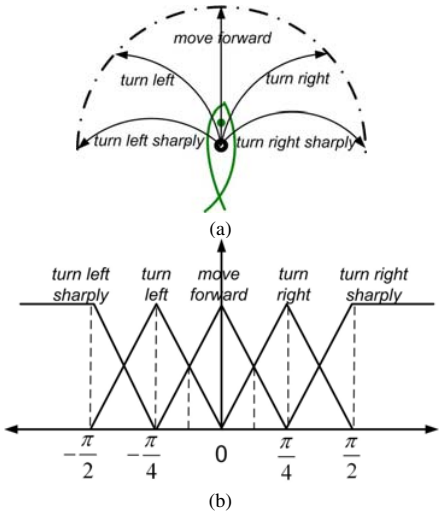


Fig. 9. Fuzzification of action space.

3) The probability of selecting a from s is determined by the Boltzman distribution function:

$$Pr(a) = \frac{exp(\frac{Q(s,a)}{T})}{\sum_{b=1}^{n} exp(\frac{Q(s,b)}{T})} \quad (7)$$

where T is the temperature parameter [9].

4) r is the reinforcement signal, which is determined by the following rules:
   - If the blocker move across between the defense and the polo, r is a positive value.
   - If the defense is forced to turn away from the polo, r is a positive value.
   - If the defense holds the polo, r is a negative value.
   - If the attacker pushes the polo into gate, r is a positive value.

After the optimal state-action pairs are obtained, we can figure out the relations of the continuous inputs and outputs. The Mamdani's minimum fuzzy implication rules [10] and the defuzzification strategy of centroid of area(COA) are used.

V. Situation-based Action Selection for Defender

For the defense player, we propose a situation-based action selection which is a hand-coded and reactive strategy. Firstly, in order to describe the situations for polo attacking, we introduce the working regions. As shown in Fig. 10, l is a line connecting the center of the polo to the goal. Let us draw a circle at the center of the polo with a radius r and then partition it into four vectors. That are four working regions: Pushing Region, Left Assistant Pushing Region, Right Assistant Pushing Region and Overshot Region. Other regions out of the circle is called Buffer Region.

- the Pushing Region (PR): this is an effective working region, since in this region the fish-like robot can push the disk to the direction of the goal and so the pushing action is effective.
- the Left Assistant Pushing Region (LAPR): this region is a semi-effective working region. The fish in this region can not directly push the disk to the desired direction, but it can help a fish in PR prevent the disk floating away.
- the Right Assistant Pushing Region (RAPR): similar to LAPR, this is also a semi-effective working region
- the Overshot Region (OR): this region is a forbidden working region. When the fish swims into this region, it will be located between the disk and the goal. In this region, the fish is forbidden to touch the disk because it will push the disk to the opposite direction to the destination. In this case, the fish should first swim around the disk and enter a effective working region and then selects suitable actions.
- the Buffer Region (BR): In this region, the fish is a little bit far from the disk, in will try to swim to an effective region (PR, LAPR or RAPR) first.

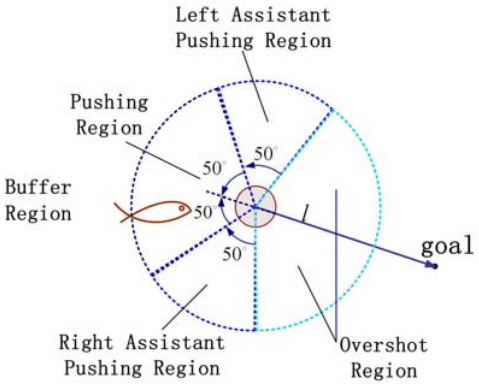


Fig. 10. Illustration of the pushing region

Based on the situations, we design the following primitive actions for the microrobot:

Action 0: This action is designed for the fish to swim from a non-pushing region to the effective working region. Basically, this is a simple $Point-To-Point(PTP)$ control.

Action 1: As depicted in Fig. 11, the first action for the fish is to swim approaching the disk and hit the disk exactly along the direction from the disk to the gate in PR. where (x_F, y_F, α) denotes the pose of the fish, (x_D, y_D) and (x_C, y_C) stand for the center of the disk and the position of the gate, β is the expected direction from the disk to the gate, l_1 indicates the expected moving direction of the disk. Considering the fish's bodylength and its inertia, We choose a point G (x_G, y_G)which is located at the extended line of l_1 as the pushing-point. l_2 is the section connecting the fish to G, and l_3 represents the perpendicular of l_1 which pass through point G.

If the pushing-point G locates between the fish and the disk, that is, $(x_D - x_F) \times (x_C - x_D) > 0$, we first define the

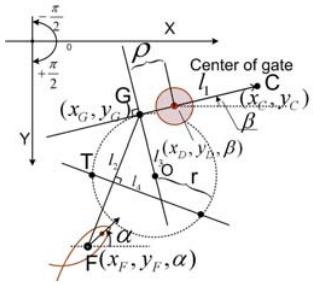


Fig. 11. The action of pushing disk along the exact direction pointing the gate

perpendicular bisector l_4 for section l_2. Then using r as radius, we make a circle C which is tangental to l_1 at point G. If the circle intersects l_4 at one point, we chose this point as a temporary goal for the fish, or if they have two intersections, we choose the point with small x-coordinates, namely T as the temporary goal. While, if the fish is far away from the disk and there is no intersection between C and l_1, G will be the temporary goal point for the fish. As the fish moves, a series of temporary goals will be obtained, which will lead the fish swim gradually to the pushing-point.

After simple geometrical analysis, the positions of intersection points can be calculated by the following equation:

$$\begin{aligned}&(x-x_D-\rho\cos\beta-r\sin\alpha)^2+(y-y_D-\rho\sin\beta+r\cos\alpha)^2=r^2\\&y-\frac{1}{2}(y_G+y^F)=\frac{x_F-x_G}{y_G-y_F}(x-\frac{1}{2}(x_G+x_F))\end{aligned} \tag{8}$$

Action 2: Although when determining the pushing-point, we give sufficient consideration for the dynamics of the fish and the difficulty when controlling it, we still can't guarantee the fish will reach its destination in the expected attitudes, especially its orientation. Once it gets to the pushing-point with large orientation error, it may possible miss the disk. In this case, we design the following action which allows the fish to push the disk by shaking head.

As shown in Fig. 12 (a), if the fish approaches the pushing-point (in a small neighbor region) and its orientation satisfies the following condition, it will take a sharp turn to the direction of the disk.

$$\begin{cases}(x_F,y_F)\in\{\|(x_G,y_G)-(x_F,y_F)\|\le\delta\}\\ \alpha\in\{|\alpha-\beta|\ge\zeta\}\end{cases} \tag{9}$$

where δ and ζ are the bounds for position error and orientation error, which are determined empirically through experiments. In our experiment, we choose $\delta=5cm$ and $\zeta=\frac{\pi}{15}$.

Action 3: This action is an assistant action, which is implemented in LAPR or RAPR. In particular, this action takes full advantage of the agility of fish's tail. Fig. 12 (b) indicates the fish pat the disk by its tail.

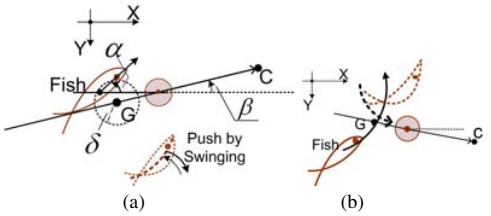


Fig. 12. The fish pushes disk by shaking(throwing) head and pushing the disk by tail.

VI. EXPERIMENTAL RESULTS

We conduct some experiments to evaluate the proposed cooperative strategy and learning approach. In all the conducted experiments, the learning process consists of a series of trials. For every trial, when the polo is pushed into the gate or the time steps number reaches 3000, the trial ends.

In Fig. 13, some typical scenarios in the experiment are presented.

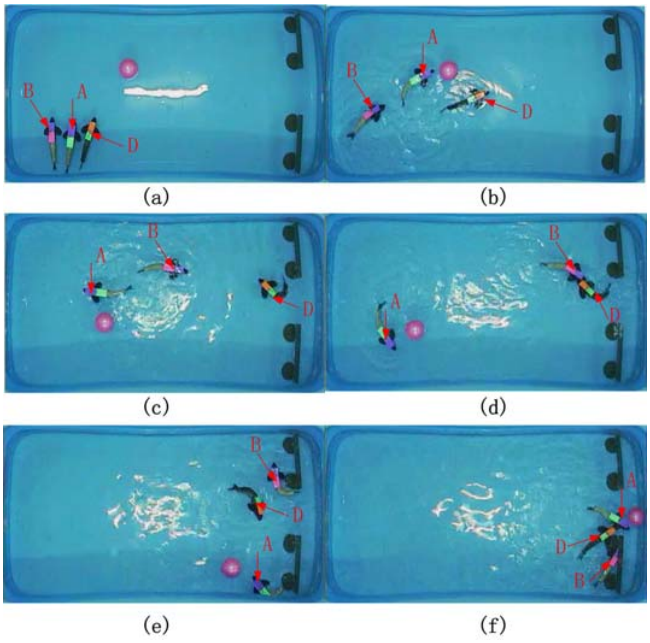


Fig. 13. Typical scenarios in one experiment. A-Attacker, B-Blocker, D-Defense.(a) The initial state of the experiment. (b) The Defense swim to the gate and defend its goal. The Attacker and Blocker start to approach the polo. (c) The Defense is just in front of the gate and watch. The Blocker overshoots the polo. (d) The Blocker blocks the Defense successfully. (e) The Blocker missed the Defense and got punishment . (f) The Attacker pushes the polo into the gate and the Offensive Side win.

Fig. 14 shows the change of the reinforcement signal of the Offensive Side in the experiment. The straight red line in the figure is called a tendency line, which is on the rise. It means during the learning process, the microrobot gets more and more positive rewards.

Fig. 15 illustrates the average time steps per trial that the Offensive Side take to push the polo into the gate. If the

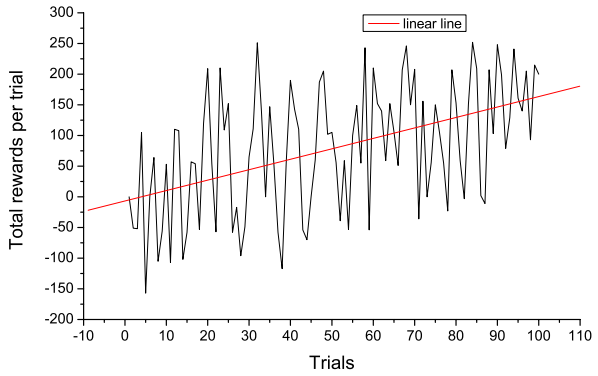


Fig. 14. the total rewards per trial versus the number of trials

Offensive Side can't manage it in a trial, we set the time steps as 3000. It is noted the tendency line descends, which means the Offensive Side take less and less time steps to win the game through learning.

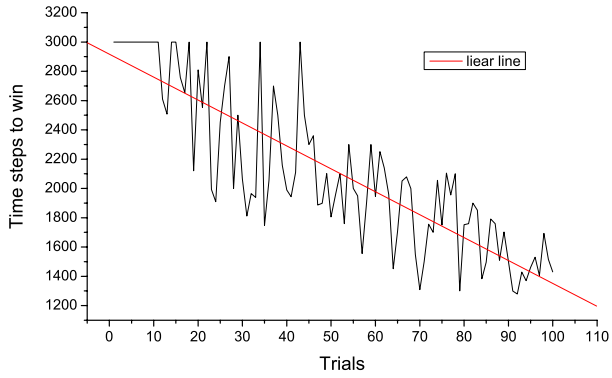


Fig. 15. The time steps to win per trial versus the number of trials

From the analysis on the results, we find the performance can hardly maintain stable at both the explorative stage and the last stage of the learning process. This might be due to the disturbance and uncertainties produced by the hydrodynamics of the environment.

VII. CONCLUSIONS

In this paper, we have presented a multiple underwater microrobot cooperative architecture and using fuzzy reinforcement to acquire the cooperative strategy in an adversarial task. Due to the complexity of the hydroenvironment, the cooperation between underwater robots is more difficult than that in ground robots. So We adopt fuzzy reinforcement learning approach to investigate this problem and aim to develop a suitable learning method to the cooperative control of underwater microtobots. Experiments verified the effectiveness of the proposed algorithms.

REFERENCES

[1] R. S. Sutton, and A. G. Barto, Reinfoecement Learning, an introduction. MIT Press, 1998.
[2] M. J. Er, and C. Deng, "Obstacle avoidance of a mobile robot using hybrid learning approach," IEEE Trans. Indus. Elec., vol. 52, no. 3, June 2005, pp. 898-905.
[3] S. Thongchai, "Behavior-based learning fuzzy rules for mobile robots," in Proc. American Control Conference, Anchorage, AK, May, 2002, pp. 995-1000.
[4] J. Yu, L. Wang, and M. Tan, "A framework for biomimetic robot fishs design and its realization," in Proc. American Control Conf. Portland, USA. pp. 1593-1598, June, 2005.
[5] J. Yu and L. Wang, "Parameter optimization of simplified propulsive model for biomimetic robot fish," in Proc. IEEE Int. Conf. Robotics and Automation. Barcelona, Spain, pp. 3317- 3322, 2005
[6] M. J. Lighthill, "Note on the swimming of slender fish," J. Fluid Mech., vol. 9, pp. 305-317, 1960.
[7] D. Barrett, M. Triantafyllou, D. K. P. Yue, M. A. Grosenbaugh, and M. J. Wolfgang,"Drag reduction in fish-like locomotion," J. Fluid Mech., vol. 392, pp. 183-212, 1999.
[8] D. Barrett, M. Grosenbaugh, and M. Triantafyllou, "The optimal control of a flexible hull robotic undersea vehicle propelled by an oscillating foil," in Proc. IEEE AUV Symp., pp. 1-9.
[9] L. P. Kaelbling, M. L. Littman, A. W. Moore. Reinforcement Learning: A survey, Journal of Artificial Intelligence Research, 1996, 4, pp. 237-285
[10] G. S. Sandhu, K. S. Rattan. Design of a Neuro Fuzzy Controller. IEEE Int. Conf. on System, Man, and Cybern., 1997, pp. 3170-3175,.
[11] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot team," IEEE Trans. Robot. Automat., vol. 14, no. 6, pp.926-939, December 1998.
[12] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer and C. J. Taylor, "A vision-based formation control framework," IEEE Trans. Robot. Automat., vol. 18, no. 5, pp. 813-825, October 2002.
[13] L. E. Parker, "ALLIANCE: an architecture for fault tolerant multirobot cooperation," IEEE Trans. Robot. Automat., vol. 14, no. 2, pp.220-240, April 1998.
[14] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in Proc. IEEE Conf. Decision and Control, Orlando, Florfida USA, December 2001, pp. 2968-2973.
[15] T. Mäenpää, A. Tikanmäki, J. Riekki and J. Röning, "A distributed architecture for executing complex tasks with multiple robots," in Proc. IEEE Int. Conf. Robotics and Automation, New Orleans, LA-April 2004, pp. 3449-3455.
[16] J. Witold, Intelligent Robotic Systems: Design, Planning and Control. New York : Kluwer Academic : Plenum, c1999.
[17] S. Whitehead, J. Karlsson, and J. Tenenberg, "Learning multiple goal behavior via task decomposition and dynamic policy merging," in Robot Learn., Norwell, MA: Kluwer, 1993.
[18] N. Ono and K. Fukomoto, "Multi-agent reinforcement learning: a modular approach," in Proc. IEEE Int. Conf. Multi-Agent Systems, 1996, pp. 252-258.
[19] K. H. Park, Y. J. Kim, and J. H. Kim, "Modular Q-learning based multiagent cooperation for robot soccer," Robot. Auton. Syst., vol. 35, pp. 109-122, 2001.